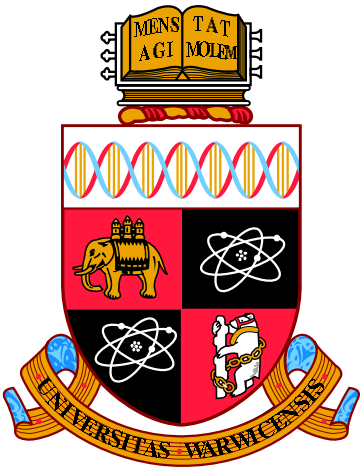


Study of the Best Linear Approximation of Nonlinear Systems with Arbitrary Inputs

by

黃　顯　鈞
Wong, Hin Kwan Roland

Thesis

Co-tutelle of

University of Warwick, U.K. & Vrije Universiteit Brussel, Belgium

for the degree of

Doctor of Philosophy

School of Engineering,
University of Warwick

Faculty of Engineering,
Vrije Universiteit Brussel

January 2013

THE UNIVERSITY OF WARWICK

Vrije Universiteit Brussel

Supervisors

Prof. Nigel G. Stocks
University of Warwick, U.K.

Prof. em. Keith R. Godfrey
University of Warwick, U.K.

Prof. Johan Schoukens
Vrije Universiteit Brussel, Belgium

Members of the Jury

Prof. Steve Vanlanduit (President)
Vrije Universiteit Brussel, Belgium

Prof. Johan Deconinck (Vice President)
Vrije Universiteit Brussel, Belgium

Prof. Gerd Vandersteen (Secretary)
Vrije Universiteit Brussel, Belgium

Prof. Jérôme Antoni
Université de Lyon, France

Prof. Tadeusz P. Dobrowiecki
Budapest University of Technology and Economics, Hungary

Dr. Ai Hui Tan
Multimedia University, Malaysia

Prof. Peter R. Jones
University of Warwick, U.K.

Contents

List of Figures

List of Tables

Acknowledgements

This thesis is dedicated to my Family, Friends and especially my Grandmother—

A dwarf standing on the shoulder of a giant...—(Bernard of Chartres, 1159)

I very much appreciate the selfless support and kind encouragements offered by my family and friends, without whom I would not be able to overcome so many failures and relish a few triumphs; for this I am eternally indebted. I am grateful to my good friends Natalie Ellis, Alice Ng, Hui Niu, Yuan Tian, Nami Morioka, Nixon Yu, Dan Zhou; my colleagues at both the University of Warwick and the Vrije Universiteit Brussel; Mr. Ho and his family at Rugby, and many others—unsung but not forgotten.

My sincerest gratitude goes to Professor Keith Godfrey and Professor Johan Schoukens, for being more than great mentors and treating me as a good friend. This endeavour would not have been possible without the technical guidance and emotional support by the two gentlemen; 'a dwarf standing on the shoulder of two giants!'—to say otherwise is an understatement. I have to thank Keith again in particular introducing and indulging me in much British culture, from visiting cathedrals, the countryside, attending opera, ballets, orchestra and musicals to punting in horse races!

I would also like to thank Dr. Ai Hui Tan for her hospitality during our visit to the Multimedia University of Malaysia and the collaboration opportunities on the benchmark studies, it was a great honour. In addition, the invaluable suggestions and encouragements offered by Professor Nigel Stocks were very much appreciated.

Last but not least, many thanks to the members of the jury committee, administrative staff and lawyers, all of whom went to extraordinary lengths to make the co-tuelle arrangements and the viva for the Ph.D. possible.

List of publications arising from this research

Journal papers

Tan, A. H., Wong, H. K. & Godfrey, K. R. (2012, May). Identification of a Wiener-Hammerstein system using an incremental nonlinear optimisation technique. *Control Engineering Practice*, *20*, 1140–1148.

Wong, H. K., Schoukens, J. & Godfrey, K. R. (2012a, March). Analysis of best linear approximation of a Wiener-Hammerstein system for arbitrary amplitude distributions. *IEEE Trans. Instrum. Meas. 61*(3), 645–654.

Wong, H. K., Schoukens, J. & Godfrey, K. R. (2013a). Design of multilevel signals for identifying the best linear approximation of nonlinear systems. *IEEE Trans. Instrum. Meas. 62*(2), 519–524.

Wong, H. K., Schoukens, J. & Godfrey, K. R. (2013b). Structured non-linear noise behaviour and the use of median averaging in non-linear systems with m-sequence inputs. *IET Control Theory and Applications*. [Accepted for Publication]. doi:10.1049/iet-cta.2012.0622

Conference papers with peer review

Wong, H. K. & Godfrey, K. R. (2010, September). A greybox approach to modelling a hyperfast switching Peltier cooling system. In *Proc. UKACC International Conference on Control*, 7th–10th September 2010 (pp. 1200–1205). Coventry, U.K.

Wong, H. K., Schoukens, J. & Godfrey, K. R. (2012b, September). Experimental verification of best linear approximation of a Wiener system for binary excita-

tions. In *Proc. UKACC International Conference on Control*, 3rd–5th September 2012 (pp. 941–946). Cardiff, U.K.

Wong, H. K., Schoukens, J. & Godfrey, K. R. (2012c, July). The use of binary sequences in determining the best linear approximation of nonlinear systems. In *Proc. 16th IFAC Symposium on System Identification*, 11th–13th July 2012 (pp. 1323–1328). Brussels, Belgium.

Declarations

This thesis was submitted to the University of Warwick and Vrije Universiteit Brussel in partial fulfilment of the requirements for the Doctor of Philosophy co-tuelle agreement between the two universities. The work presented here is my own, except where specifically stated otherwise. The work was performed both in the School of Engineering at the University of Warwick under the supervision of Professor Nigel G. Stocks and Professor Emeritus Keith R. Godfrey, and in the Department of Fundamental Electricity and Instrumentation, Faculty of Engineering at the Vrije Universiteit Brussel under the supervision of Professor Johan Schoukens.

The benchmark study documented in Chapter 7 was a collaboration with Dr. Tan, Ai Hui of the Multimedia University, Cyberjaya, Malaysia.

Abstract

System identification is the art of modelling of a process (physical, biological, etc.) or to predict its behaviour or output when the environment condition or parameter changes. One is modelling the input-output relationship of a system, for example, linking temperature of a greenhouse (output) to the sunlight intensity (input), power of a car engine (output) with fuel injection rate (input). In linear systems, changing an input parameter will result in a proportional increase in the system output. This is not the case in a nonlinear system. Linear system identification has been extensively studied, more so than nonlinear system identification. Since most systems are nonlinear to some extent, there is significant interest in this topic as industrial processes become more and more complex.

In a linear dynamical system, knowing the impulse response function of a system will allow one to predict the output given any input. For nonlinear systems this is not the case. If advanced theory is not available, it is possible to approximate a nonlinear system by a linear one. One tool is the Best Linear Approximation (Bla), which is an impulse response function of a linear system that minimises the output differences between its nonlinear counterparts for a given class of input. The Bla is often the starting point for modelling a nonlinear system. There is extensive literature on the Bla obtained from input signals with a Gaussian probability density function (p.d.f.), but there has been very little for other kinds of inputs. A Bla estimated from Gaussian inputs is useful in decoupling the linear dynamics from the nonlinearity, and in initialisation of parameterised models. As Gaussian inputs are not always practical to be introduced as excitations, it is important to investigate the dependence of the Bla on the amplitude distribution in more detail. This thesis studies the behaviour of the Bla with regards to other types of signals, and in particular, binary sequences where a signal takes only two levels. Such an input is valuable in many practical situations, for example where the input actuator is a switch or a valve and hence can only be turned either on or off.

While it is known in the literature that the Bla depends on the amplitude distribution of the input, as far as the author is aware, there is a lack of comprehensive theoretical study on this topic. In this thesis, the Blas of discrete-time time-invariant nonlinear systems are studied theoretically for white inputs with an

arbitrary amplitude distribution, including Gaussian and binary sequences. In doing so, the thesis offers answers to fundamental questions of interest to system engineers, for example: 1) How the amplitude distribution of the input and the system dynamics affect the BLA? 2) How does one quantify the difference between the BLA obtained from a Gaussian input and that obtained from an arbitrary input? 3) Is the difference (if any) negligible? 4) What can be done in terms of experiment design to minimise such difference?

To answer these questions, the theoretical expressions for the BLA have been developed for both Wiener-Hammerstein (WH) systems and the more general Volterra systems. The theory for the WH case has been verified by simulation and physical experiments in Chapter 3 and Chapter 6 respectively. It is shown in Chapter 3 that the difference between the Gaussian and non-Gaussian BLA's depends on the system memory as well as the higher order moments of the non-Gaussian input. To quantify this difference, a measure called the Discrepancy Factor—a measure of relative error, was developed. It has been shown that when the system memory is short, the discrepancy can be as high as 44.4%, which is not negligible. This justifies the need for a method to decrease such discrepancy. One method is to design a random multilevel sequence for Gaussianity with respect to its higher order moments, and this is discussed in Chapter 5.

When estimating the BLA even in the absence of environment and measurement noise, the nonlinearity inevitably introduces nonlinear distortions—deviations from the BLA specific to the realisation of input used. This also explains why more than one realisation of input and averaging is required to obtain a good estimate of the BLA. It is observed that with a specific class of pseudorandom binary sequence (PRBS), called the maximum length binary sequence (MLBS or the m-sequence), the nonlinear distortions appear structured in the time domain. Chapter 4 illustrates a simple and computationally inexpensive method to take advantage this structure to obtain better estimates of the BLA—by replacing mean averaging by median averaging.

Lastly, Chapters 7 and 8 document two independent benchmark studies separate from the main theoretical work of the thesis. The benchmark in Chapter 7 is concerned with the modelling of an electrical WH system proposed in a special session of the 15$^{\text{th}}$ International Federation of Automatic Control (IFAC) Symposium on System Identification (SYSID) 2009 (Schoukens, Suykens & Ljung, 2009). Chapter 8 is concerned with the modelling of a 'hyperfast' Peltier cooling system first proposed in the U.K. Automatic Control Council (UKACC) International Conference on Control, 2010 (Control 2010).

Acronyms and Abbreviations

† Bold page numbers point to pages in which terms are defined or explained in the text.

Acronym	Abbreviation	Pages†
ADC	Analogue-to-digital converter	98
ARE	Asymptotic relative (statistical) efficiency	75–77, 82
AWGN	Additive white Gaussian noise	**15**, 58, 62, 65, 78, 82
BLA	Best Linear Approximation	ix, 4–9, 18, 25, 27, **28**, 29–31, 33–39, 41–49, 51, 53, 55–58, 60, 63, 65, 66, 68–74, 76, 78–80, 83, 86, 87, 91, 93–95, 97, 99, 101, 103, 104, 106–111, 113–116, 123, 132, 145, 147–151, 165, 167, 169–172, 174, 176, 184
CF	Crest factor	14, 18, 95, 96
DF	Discrepancy Factor	7, 9, **47**, 48–53, 55, 56, 86, 87, 91, 92, 148, 150, 151
DFT	Discrete Fourier transform	**13**, 14–16, 18, 19, 21, 23, 24, 59, 69, 73, 78, 129
DIRBS	Discrete-interval Random Binary Sequence	**15**, 16, 17, 32, 33, 38, 49, 61–63, 65, 66, 69, 73, 77, 78, 82–84
DTFT	Discrete-time Fourier transform	**12, 13**
FFT	Fast Fourier transform	20
FIR	Finite impulse response	35, 39, 65, 66, 69, 77, 109, 116, 138

Acronym	Abbreviation	Pages†
Frf	Frequency response function	4, 13, 14, 27, 39, 69, 116
Ft	Fourier transform	4, 11, **12**, **13**, 106
Hlle	Hodges-Lehmann location estimator	77, 78, 82
Iir	Infinite impulse response	35, 39, 77, 80, 84, 109, 116
Irbs	Inverse-repeat Binary Sequence	14, **24**, 63, 65, 66, 68, 83
Irf	Impulse response function	4, 13, 14, 27, 28, 35, 38, 41, 44–46, 49, 50, 52, 53, 55, 56, 65, 69–71, 77, 78, 80, 82, 83, 106, 109, 172, 175–177
Lfsr	Linear feedback shift registers	19–21, 23, 24, 163
Mae	Mean absolute error	141, 145
Miso	Multiple-input single-output	135
Mlbs	Maximum Length Binary Sequence	7, 9, **18**, 19–25, 32, 33, 38, 39, 60, 61, 63, 65, 66, 68, 69, 71, 72, 74, 78–80, 82–84, 99, 100, 103–105, 107, 108, 149–151, 163, 181
Mle	Maximum likelihood estimator	82
Mse	Mean squared error	62, 66, 78, 80–83, 112, 141, 145
Oe	Output-error (model/framework)	57, 62
p.d.f.	Probability density function	xxiv, 5, 15, 17, 18, 25, 47, 75, 86
Pi	Periodicity-invariant (system)	**11**, 12, 15, 28, 34, 35, 165
Pips	Performance Index for Perturbation Sequences	14, 95, 96
p.m.f.	Probability mass function	5, 25, 47, 86, 88–90, 95
Prbs	Pseudorandom binary sequence	**18**, 60, 85
Prms	Pseudorandom multilevel sequence	25, 85
rms	Root mean squared	33, 100, 104, 106, 113, 114, 116, 182
Rse	Relative statistical efficiency	**74**, 75, 76, 82
Siso	Single-input single-output	2, 33, 43, 136
Snr	Signal-to-noise ratio	9, 14, 16, 18, 62, 65, 77, 79, 81–83, 95, 149, 151

Acronym	Abbreviation	Pages†
WH	Wiener-Hammerstein (system)	7, 31, 33, 34, 43, 56, 86, 111, 112, 115, 116, 125, 132, 133, 147
XOR	EXclusive-OR logic gate	20, 21, 24
ZOH	Zero-order hold	16, 23, 53, 99, 101, 102, 112

Glossaries

Blackbox model — In system identification, the term *blackbox* modelling refers to the process of modelling a system through non-parametric techniques, without knowledge of physical inner workings of the system, resulting in an *empirical* model. 8, 112, 136, 138, 145

dc stands for direct current in electrical engineering. Regardless of the physical nature of a signal (electrical or otherwise), the terms dc offset, dc bias, dc term or dc component are all used widely in signal processing to refer to the mean value of a waveform in the time domain or the zero frequency component in the frequency domain. viii, 15, 16, 19, 23, 24, 34, 38, 39, 43, 66, 125

ELiS stands for "Estimator for Linear Systems", a program in the Frequency Domain System Identification (*FDIDENT*) toolbox for MATLAB. It implements an iterative weighted nonlinear least squares procedure that fits parametric models to non-parametric transfer characteristics of dynamical systems (Kollár, 1994). 49, 58, 91, 104, 108, 113, 114, 120, 125, 127

Empirical model — *see* blackbox model

GALOIS is a program to generate pseudorandom signals including maximum-length sequences with various number of levels Barker (2008), Godfrey, Tan, Barker and Chong (2005). The signals can be designed to have desirable properties for dynamic system testing. 24, 85

Gaussianity is a qualitative measure of how close the statistical distribution of a random variable is to a true Gaussian distribution, also known as a normal distribution. xviii, 7, 9, 18, 86, 93, 96

Greybox model — In system identification, a *greybox* modelling approach is a mixture between whitebox and blackbox modelling. 136, *see also* blackbox model & whitebox model

MATLAB (MATrix LABoratory) is a widely used scientific numerical computation environment developed by MathWorks with its own programming language. The platform is optimised for matrix and vector based calculations and offer a wide range of tools for the manipulation, processing and visualisation of data. xxii, xxiv, 17, 24, 38, 58, 62, 63, 65, 77, 93, 98, 104, 106, 113, 116, 121, 127, 129, 137, 146, 181–183

Mechanistic model — *see* whitebox model

Moments — see definition on p. 32. xxiii

Non-parametric model is a data-driven model without *a priori* specification of a model structure; instead the structure is determined entirely from the data. Essentially the parameters are not fixed and typically grow in size according to the amount of data available see. 30, 39, 49, 58, 68, 91, 93, 104, 106–108, 136

Parametric model is a problem-driven model with a fixed model structure and a predetermined number of parameters. A good model can capture maximal useful information from the data used to derive it with only a few number of parameters. 49, 91, 104–108, *See also* non-parametric model

Probability density function (p.d.f.) — In probability statistics, a p.d.f. of a continuous random variable is a density function of x such that, when integrated over a range of x to give the area under the curve, the area describes the relative likelihood for finding this random variable to lie within that range at any given time. The p.d.f. defines the amplitude distribution and the moments for the random variable. *See also* moments & probability mass function (p.m.f.)

Probability mass function (p.m.f.) — In probability statistics, a p.m.f. of a *discrete* random variable is a discrete function of discrete variable x that describes the relative likelihood for finding this random variable to take on a given value x at any given time. The p.m.f. defines the distribution and the moments for the random variable. The p.m.f. for a discrete variable is the discrete analogue

of the continuous probability density function (p.d.f.). *See also* moments & probability density function (p.d.f.)

prs stands for "pseudorandom sequences" and is a collection of programs written for MATLAB software to generate maximum-length sequences and primitive polynomials in a convenient package (Tan & Godfrey, 2002). 24, 85, 181

Simulink is a commercial tool developed by MathWorks with a tight integration with MATLAB software that offers a block-oriented multi-domain programming approach to modelling, stimulating and analysing dynamic systems (Krauss, Shure & Little, 1994). 137, 138, 146, *see also* MATLAB

Uniform distribution — In probability statistics, a random variable x with a uniform distribution has equal relative likelihood (or probability) of taking any permissible values. The uniform distribution may be discrete or continuous and is usually defined by a range, for example, $a \leq x \leq b$. 17, 93, *See also* moments, probability density function (p.d.f.) & probability mass function (p.m.f.)

Whitebox model — In system identification, the term *whitebox* modelling refers to the process of modelling a system through first principles, laws of physics and explicit assumed relationships between the input and output through prior knowledge of the system, resulting in a *mechanistic* model. 8, 136, 145

Mathematical notations and typesetting conventions

Unless otherwise specified, notations for symbols and mathematics are as follows:

Table 2: Table of notations and for symbols and mathematics

Symbol	Description
$\approx$	... approximately equals to ...
$\equiv$	... is a mathematical identity of ...
$\triangleq$	... by definition equals to ...
$\forall$	... for all ...
μ	Arithmetic mean
σ	Standard deviation
e	Euler's number (Napier's constant)
f, ω	Frequency variables (Hz, $\mathrm{rad\,s^{-1}}$)
$\mathfrak{f}(x)$	Generic function on x
$\mathfrak{f}(x) \mapsto y$	$\mathfrak{f}(x)$ is defined such that x maps to y
π	Ratio of a circle's circumference to its diameter
j	Imaginary number, equals $\sqrt{-1}$
$\mathfrak{f}(\theta') = \arg\min_{\theta} \mathfrak{f}(\theta)$	Specific value $\theta = \theta'$ that minimises $\mathfrak{f}(\theta)$ resulting in the minimised function of $\mathfrak{f}(\theta')$
$g_{\mathrm{BLA}}(t)$	BLA (time domain)
$G_{\mathrm{BLA}}(\mathrm{j}\omega)$	BLA (frequency domain)
T	Sampling interval or bit-interval
$u(t)$	Continuous input signal u at time t
$u[k]$	Sampled input signal u at sample k, k counts from zero, i.e. $u[0] = u(0 \leq t < T)$

continued on next page

continued from previous page

Symbol	Description
$y(t), y[k]$	Similar definitions to $u(t)$ and $u[k]$ respectively but for the output signal y instead of u
$\mathcal{F}\{x\}$	The DFT of x
$\mathcal{F}^{-1}\{X\}$	The inverse DFT of X
$\mathcal{O}$	Big-O notation, describes the limiting behaviour of a function
$\mathcal{N}(\mu, \sigma^2)$	Gaussian (normal) distribution of mean μ and variance σ^2
$\mathcal{U}(a, b)$	Continuous uniform distribution between a and b
$\mathcal{U}_n$	Symmetrical discrete uniform distribution with n levels
$x \sim \mathcal{N}$	Random variable x follows a normal distribution
$x \in \boldsymbol{v}$	x belongs to, or is a member of, $\boldsymbol{v}$
$x!$	The factorial of x
$x!!$	The double factorial of x
$\mathrm{E}[\![x]\!]$	The expected value of the random variable x
$\{x\}$	A set of x
$\langle u \rangle$	A sequence of u
$\boldsymbol{v}^{\mathrm{T}}$	The transpose of vector $\boldsymbol{v}$
R_{uu}	The autocorrelation function of the signal u
R_{yu}	The cross-correlation function between the signals y and u
$\mathbb{N}$	Set of natural numbers excluding zero
$\mathbb{N}_0$	Set of natural numbers including zero
$\lvert x \rvert$	Absolute value, magnitude or cardinality (set theory) of x
$\boldsymbol{a} \backslash \boldsymbol{b}$	a complement b (set theory)
$\boldsymbol{a} \subset \boldsymbol{b}$	a is a proper subset of b (set theory)
$\boldsymbol{a} \subseteq \boldsymbol{b}$	a is a subset of b (set theory)
$\mathbb{1}_{\boldsymbol{A}}(x)$	The multiplicity function for multiset $\boldsymbol{A}$ operating on element x (set theory)
s	s-domain operator
z	z-domain operator
$\mathfrak{R}$	Real number axis
$\mathfrak{I}$	Imaginary number axis
$x\vert_{a=1}$	Expression x applies given condition $a = 1$ is satisfied
$\mathfrak{D}$	Discrepancy Factor, see Section 3.5
$\mathfrak{M}_n$	Notation for Moments, see Section 3.2

The conventions for typesetting are given are given in the following table:

Table 3: Table of typesetting conventions

Typesetting	Example	Description
De-italicised Roman	$\mathrm{f}, \mathrm{E}, z^{-1}$	Functions or domain operators
Italic	x	Variables
Uppercase italic	X	Complex frequency domain variables or scalar constants
(Subscripts)	k_{a}	Descriptive subscripts are normally upright, except when the subscript itself is an index variable
Bold italic lowercase	$\boldsymbol{\xi}, \boldsymbol{v}$	Sets or vectors
Bold italic uppercase	$\boldsymbol{\Xi}, \boldsymbol{A}$	Multisets or matrices
Calligraphy style	$\mathcal{P}$	Mathematical classes (collection of sets or multisets) or statistical distributions
Blackboard bold style	$\mathbb{Z}$	Number sets or speciality functions
Fraktur script	$\mathfrak{D}, \mathfrak{M}$	Special entities
Italicised text	*odd*	Emphasis
Slanted text	*ID 30*	Object labels

Chapter 1

Introduction

1.1 Background of System Identification

The field of system identification and modelling is relatively modern. Before its generalisation, systems science was born out of necessity from several different fields, for example, biomedicine, neuroscience, meteorology, signal processing, communications, geology and acoustics. They all share a common goal: to create mathematical models capable of encapsulating some real processes and subsequently allow one to make prediction and, depending on application, control or influence the process to give certain desirable outcome.

1.1.1 Systems and models

In the broad sense, a system is a collection of interacting entities. For example, the central nervous system is a network of interacting neurons, nerve cells as sensors and muscle cells, amongst others, as actuators. Another example is the orbits of planets around a star, which are governed by the physical law of general relativity. The former example has clear definition of inputs, measured by some sensors, and outputs, articulated by some actuators; the latter however, is less clear and depends on definition of the problem in hand. In the star system example, an astronomer may define the inputs of a star system being the masses, positions, velocities of the celestial bodies and define the output as the geometry of the orbits; understanding the model will allow the astronomer to make predictions of the output, i.e. the orbital information.

Similarly, in system and control theory, a system is a process with single or multiple inputs directly influencing or determining the quantity of a single or some combinations of outputs. System and control engineers sought to understand the relationships between these inputs and outputs, and therefore to make predictions or devise the best way to manipulate the inputs in order to attain certain desired output behaviour. Many systems studied are dynamical—their outputs at any given time simultaneously depend on more than one past instants of inputs, outputs, or a combination of both. In other words, the outputs of the immediate future are not just determined by the inputs or outputs at this very current moment, but also the values in past moments. Exhibiting dynamics is also called having memory effects. In the frequency domain a dynamical system is characterised by having non-constant magnitude and phase frequency responses. A system without dynamics is called a static system.

A model of a system is a mathematical description of the relationships governing the inputs and outputs. Such a model may contain a complete description or an approximation of the system. Unless full physical knowledge and the corresponding law of physics are well understood, it is usually not possible to devise a complete description. To be able to understand the relationship between the inputs and outputs, it is necessarily to apply techniques from the disciplines of system identification and modelling. System *identification* is the art of extracting information and gathering data efficiently and accurately from the system in question with certain user-specified goals. Sometimes it is possible to take control of an input, which simplifies the process. In such cases, the choice of the types and properties of input signals is important to the quality of information extracted. With data gathered, one can then start building models of the system. System *modelling*, on the other hand, is the art of choosing, developing and validating models that give the most accurate and precise description of the real system based on the finite measurement data obtained from system identification. A good model allows one to predict the output or outputs of a system given some arbitrary inputs, or at the very least, normal operating inputs expected for the real operation of the system. Once a model has been derived and validated, the foundation is set for control engineers to devise optimised control schemes to influence a system to achieve certain desired behaviour.

In this thesis, for simplicity, only dynamical systems with a single input and a single output are considered, these systems are referred to as single-input single-output (Siso) systems. The systems considered are also time-invariant. This is to say, the system dynamics, structure and behavior remain unchanged with time so

that under the same input and given the same initial conditions, the system always generates the same output.

1.1.2 Linear and nonlinear systems

Linear models are well documented and commonly used, even when the underlying system is nonlinear (Ljung, 1999, tbl. 4.1). The concept of linearisation and proportionality is well known and used in a wide range of fields. Some relationships in nature are predominantly linear, some are very nonlinear and there are those in between. All 'linear' systems are nonlinear to a certain extent—there are no true linear relationships, as ultimately the linearity breaks down at some extreme points due to changing law of physics or when external factors are introduced and starting to dominate. Even with moderately nonlinear systems, linear models of sufficient quality can be obtained given a small enough input domain or operation range. Mathematicians, scientists and engineers prefer to apply linear approximations because their simplicity opens the door for closed-form algebraic study and allows models to be devised with minimal effort and cost.

Definition A nonlinear system is a system where the input-output relationship(s) does not obey the principle of superposition.

An example of a nonlinear system is the human auditory system—doubling the power output of a loudspeaker does not double the perceived loudness. It also has dynamics—different loudness is also perceived for tones with different frequencies. Note that in this thesis, the term *nonlinear system* is used for a dynamical nonlinear system while the term *static nonlinearity* is used to refer a static nonlinear entity.

1.1.3 Impulse and impulse response

Before the existence of modern system identification techniques, a frequently used excitation signal called an *impulse* function (more formally the Dirac delta function) has been used throughout human history, even before its mathematical understanding was developed. Since the ancient times, people have found that one can examine the quality, material, geometry or mass of an object by striking it with a short-lasting impact. For example, knocking on a hollow object produces a distinctively different sound than that produced by knocking on a solid object. By listening to the knock, one performs an 'identification' with one's ears and brain—together make a surprisingly good biological audio frequency spectrum analyser! Many musical instruments also rely on impulses. For instance, pianists play sound by hitting keys, which causes

tensioned wires in the piano to be struck by hammers. Each key corresponds to a different wire that was taut with a specifically tuned tension force (or made with an entirely different metal), in order to have its resonant frequency corresponding to a particular musical note. The strike of the hammer is effectively an impulse that causes the struck wire to vibrate and resonate at its natural frequency. This vibration interacts with the sound board of the piano and the surrounding air to produce the musical note as a perceivable sound. An example of successful application of impulses in a system identification problem is in the analysis of the acoustics of a concert hall though gunshots and even balloon bursts (Sumarac-Pavlovic, Mijic & Kurtovic, 2008).

With an impulse as excitation, it is possible to extract information of the dynamics from a system. The response as a function of time of a linear dynamical time-invariant system to an applied impulse is called the impulse response function (IRF), and is sufficient to describe the dynamics (but not the inner-structure) of a linear time-invariant dynamical system. Knowing the IRF allows one to predict the output of the system under any arbitrary inputs (not outside operating ranges that may induce nonlinear effects). Because of this, the objective of many system identification exercises is to obtain the IRF. The frequency response function (FRF) is the analogue of the IRF in the frequency domain. Through the use of Fourier transform (FT) there is a one-to-one mapping between the frequency domain and the time domain (see Section 2.2).

For nonlinear systems, the closest analogue of an IRF is the Volterra kernel (Volterra, 1930/2005), akin to a multidimensional IRF. This will be discussed in more detail in Section 3.4. However, it is possible to devise an IRF to represent a linear model that behaves, as close as possible, to a nonlinear system. One such tool is the Best Linear Approximation (BLA), introduced in the next section and discussed in more technical detail in Section 3.1.

1.1.4 The Best Linear Approximation (BLA)

To devise a model to fit the relationship of some data variates, firstly one has to define a measure of error between the model prediction and the true measured data, called the cost function. A popular choice is the least squares cost, where the total error is defined as the total squared differences between the model prediction and the true measurement. A model where this total error is minimised is called the least squares model. For example, in statistics, the ordinary least squares linear regression is a type of least squares model.

In dynamical systems, the least squares linear time-invariant model of a nonlinear time-invariant system is called the Best Linear Approximation (BLA). It was founded on the basis of the Bussgang theorem (Bussgang, 1952) and it has several important properties under the influence of input signals having Gaussian p.d.f's.: firstly, the transfer characteristics of a block-structured nonlinear system is the combined dynamics multiplied by certain constant factor depending on the nonlinearity (for example, see Section 3.3); and secondly, the BLA of a static nonlinearity is static. In this thesis, for convenience, the BLA obtained from signals bearing a Gaussian p.d.f. (or p.m.f. for discrete sequences) shall be called the *Gaussian BLA*.

However, the aforementioned properties do not apply if the inputs to the nonlinear system do not have a Gaussian distribution. Example 1.1 of Enqvist (2005) shows that the BLA of a static nonlinearity for non-Gaussian inputs is not necessarily static. Further, example 1.2 from the same source shows that the BLA can be different for two inputs despite sharing an equal colour (or power spectrum). A motivation example illustrating the dependence of the BLA on the amplitude distribution of the input signals is given as follows:

Example Consider a simple pure cubic nonlinearity such that the output y is related to the input u by $y = u^3$, and hence there are no dynamics. Three types of zero-mean inputs: Gaussian, symmetrical (about zero) binary (2-level) sequences and symmetrical ternary (3-level) sequences are used to identify the BLA of the cubic nonlinearity. The binary and ternary sequences have their levels uniformly distributed, i.e. all their levels have equal probability of appearance, the value of which is $1/2$ for the binary levels and $1/3$ for the ternary levels. The power of all three sequences is normalised to unity. As a result, the variance of the Gaussian input is also unity. For the other two sequences, the power normalisation combined with the symmetry dictate that the binary sequence to comprise levels $+1$ and -1 and the ternary sequence to comprise levels of $+\sqrt{1.5}$, 0 and $-\sqrt{1.5}$ (see methods in Chapter 5 for the derivation). The results are shown in Fig. 1.1. It can be seen that the BLAs are indeed different for all three types of inputs. This is understandable as the BLAs are optimised to approximate the nonlinear cubic relationship to their best ability (in terms of least squared errors) within the range of input amplitudes of their corresponding input signal. This means that for the binary case, the BLA is exactly described by $y = u$, as the error between $y = u$ and $y = u^3$ is exactly zero given input levels $u \in \{+1, -1\}$. For the Gaussian case the BLA is $y = 3u$ while for the ternary case the BLA is $y = 1.5u$. The theoretical steps to derive these BLAs shall not be discussed here as the theory is not introduced until Chapter 3. Nonetheless, note that the coefficient of u of each BLA in this example is directly

equal to the 4^{th} order moments of the corresponding input. The concept of moments will be introduced in Section 3.2 and within which, Table 3.1 contains the central moments for the three types of sequences used in this example.

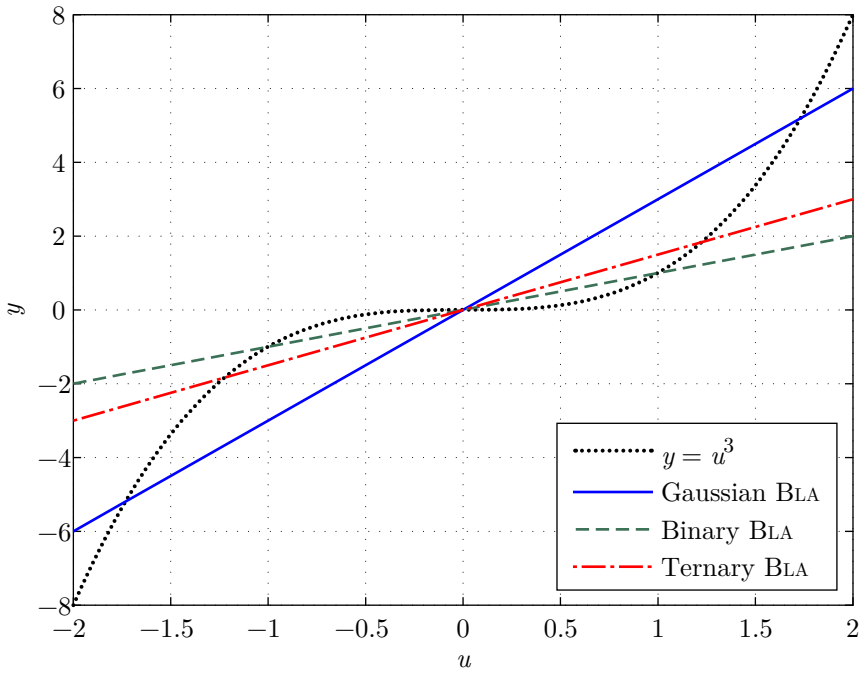


Figure 1.1: The BLAs of a static pure cubic nonlinearity identified by three types of input signals

As mentioned before, Gaussian BLAs have desirable and well known properties. However, in some systems, one cannot expect the inputs to have a Gaussian distribution—the flow rate of a fluid controlled by an on-off valve for example, is more binary in nature; this sets the motivation of the thesis. Questions arise, for example, what happens if a binary excitation is used instead of a Gaussian noise excitation? Will the results significantly change, or will the differences be sufficiently small that, for practical purposes, they can be ignored? Is it possible to quantify the differences between non-Gaussian and Gaussian excitations on the basis of the non-Gaussian measurement results? These questions will be addressed in Chapter 3.

1.2 Thesis outline

This thesis is split into two parts containing nine chapters in total. The first six chapters revolve around the main topic on the theory of the non-Gaussian BLA.

Chapters 7 and 8 are both standalone chapters each documenting a benchmark study. The last chapter contains general conclusions, a discussion on future work and a summary of contributions. Details of each chapter in this thesis are listed as follows:

Chapter 2 introduces the various periodic input sequences used throughout this thesis with discussions on their properties and how they are generated.

The first part of Chapter 3 gives a formal definition of the BLA and shows that it is possible to derive closed-form expressions for the BLAs obtained from signals with arbitrary amplitude distributions for any time-invariant discrete-time systems; within which two types of systems are shown: firstly the block-structured Wiener-Hammerstein (WH) system in Section 3.3 and secondly for the more general Volterra systems in Section 3.4. The second part of Chapter 3 introduces a measure to quantify the difference called the Discrepancy Factor (DF) in Section 3.5. This answers the question on how much the BLA deviates from the Gaussian obtained counterpart and it is shown that systems with shorter memory, in general, have higher DFs.

The first part of Chapter 4 introduces a method for identifying the BLA and discusses the behaviour of exogenous environment (and measurement) noise and nonlinear distortions arising from the nonlinearities, with respect to different available input sequences introduced in Section 2.3. The second part from Section 4.5 onwards documents a method to improve the estimates of the BLA through using alternative averaging schemes rather than the conventional mean function, by using a special property of Maximum Length Binary Sequence (MLBS) inputs resulting in structured nonlinear distortions. The results show such a method is simple to apply and effective in low exogenous noise scenario.

Chapter 5 illustrates a method to design discrete-interval random multilevel sequences for Gaussianity. Gaussian BLA has well documented properties, one of which is the ability to decouple the dynamics from the static nonlinear contributions. Some problems prohibit the use of continuous amplitude levels, examples of which are given in the beginning of the chapter. In such situations, optimising multilevel sequences for Gaussianity is one method to obtain an unbiased Gaussian BLA.

Chapter 6 details an experiment based on an electrical circuit in a Wiener system configuration conducted to verify the discrete-time BLA theory developed in Chapter 3, in which only simulated experiment was conducted. The experiment results successfully verified the theory in a laboratory setting.

Chapter 7 proposes a solution to the benchmark problem with a Wiener-Hammerstein electrical system published in Schoukens, Suykens and Ljung (2009).

The content of this chapter is based on Tan, Wong and Godfrey (2012). The chapter shows a utilisation of one property of the Gaussian BLA—the BLA is proportional to the combined linear dynamics of the system. This allows a brute-force exhaustive search of all poles and zero combinations resulting in two physically realisable filters for the first and second linearities of a Wiener-Hammerstein system. The iterative algorithm proposed also models the static nonlinearity using a polynomial or a piecewise polynomial function. The parameters are tuned in multiple stages, first for linearity assignment, then to nonlinearity, to minimise the number of parameters the optimisation algorithm has to tackle at any stage. The polynomial degrees are increased one degree at a time to minimise the chance of the optimisation being stuck in a local minimum.

Chapter 8 documents a modelling exercise on a 'hyperfast' Peltier cooling system assembly (Cham, Tan & Tan, 2010). The solution encompasses a mixture of blackbox and whitebox modelling approaches, incorporating as much physical and structural knowledge of the system as possible and without relying on numerical non-parametric optimisation methods. The solution shows that even a simple, naïve approach that does not use established sophisticated system identification methods, can yield models with acceptable performance given enough prior physical knowledge. This piece of work was based on Wong and Godfrey (2010).

1.3 Contributions

Much of the research documented in this thesis has previously been published in journal or conference publications; a list of which is available on p. xiv. Unless otherwise specified or referenced, all the work are performed by the author under the guidance of the supervisors named on p. ii with the exception of Chapter 7 which was part of a collaboration (see p. xvi). Major contributions arising from the research work performed over the course of this Ph.D. degree are highlighted in this section as follows:

The objective of this thesis is mainly concerned with the use of non-Gaussian signal in identifying the BLA of the system. While the dependence of the BLA on the input amplitude distribution is well known in the literature, the work in Chapter 3, first published in Wong, Schoukens and Godfrey (2012a), tackles some questions relevant to the system identification community. To the best of the author's knowledge, the effect of non-Gaussianity of inputs to the BLA of a time-invariant discrete-time nonlinear system has not been investigated in detail, and this thesis is able to provide closed-form algebraic expressions for the BLA of generalised discrete-time Volterra

systems with arbitrary input distributions. This answers the question made in the beginning of the chapter: 'What happens if a binary excitation is used instead of a Gaussian noise excitation?'. Another question is: 'Will the results significantly change, or will the differences be sufficiently small that, for practical purposes, they can be ignored?'. The simulation experiments performed for a Wiener system with a) cubic (Section 3.3.1) and b) quintic (Section 3.3.2) nonlinearity have shown that the Gaussian BLA and the BLA obtained from other input signals can have a significant difference between them. Finally, to answer: 'Is it possible to quantify the differences between non-Gaussian and Gaussian excitations on the basis of the non-Gaussian measurement results?', a measure called the Discrepancy Factor (DF) was developed in Section 3.5 to quantify the difference.

The first part of Chapter 4 discusses the behaviour of noise and nonlinear distortions and how it affects the choice of input signal. This piece of work was also published in Wong, Schoukens and Godfrey (2012c). In the second part, significant improvement to the quality of estimate of the BLA of a nonlinear system can be made when the input sequences used were Maximum Length Binary Sequences (MLBS's), giving rise to nonlinear distortions having a certain structure unique to MLBS's. While this structured behaviour is known in the literature, see for example, Godfrey and Moore (1974), Vanderkooy (1994), the structured nonlinear noise has been treated as a negative aspect rather than an opportunity. In this thesis, it was found that the use of alternative averaging schemes rather than the traditional mean-based averaging can offer significant improvement in the BLA estimate under moderate to high signal-to-noise ratio (SNR) due to the structured nonlinear distortions. The findings were submitted as Wong, Schoukens and Godfrey (2013b), which has been accepted for publication.

Another notable contribution involves the physical experiment performed to verify the theory developed in Chapter 3, documented in Chapter 6 of this thesis and published in Wong, Schoukens and Godfrey (2012b). Here the experiment showed good agreement between the theory and practice.

Furthermore, noting the dependence of the BLA on Gaussianity of the input signal, the author investigated the merits in designing discrete-time discrete-level sequences up to 5 levels (quinary) to match the higher order statistics to a discrete-time Gaussian sequence. Chapter 5 has shown that it is possible to, depending on the degree of nonlinearity, minimise or completely eliminate the bias of the estimated BLA with respect to the Gaussian BLA. This piece of research was also published to Wong, Schoukens and Godfrey (2013a) and was presented at the 31st Benelux Meeting on Systems and Control, held on the 27th–29th, March, 2012 in Heijen, the

Netherlands.

Lastly, the benchmark Chapters of 7 and 8 document solutions to benchmark problems previously proposed in two separate conferences. The benchmark studies provided valuable training opportunities for the author in system identification.

Chapter 2

Periodic Input Sequences

Periodic sequences are widely used in system identification. They offer several advantages over aperiodic sequences in, for instance, eliminating spectral leakage (see Section 2.2), the ability to perform inter-period averaging to drive down the effect of exogenous environment noise (see Section 4.2) and allowing the user to compare the level of nonlinear distortions to that of noise (Pintelon & Schoukens, 2012). Several types of periodic input sequences used throughout this thesis are described in this chapter. Before these sequences are introduced in Section 2.3, two concepts first explained: periodicity-invariant (Pi) property in Section 2.1 and the various Fourier transforms (Fts) in Section 2.2.

2.1 Periodicity-invariance (Pi)

Definition A *nonlinear* system is designated as periodicity-invariant (Pi) when excited by any periodic input u with period N; the noise-free output y would always share the same period N.

This is the case if the spectral harmonic frequencies of the discrete output spectrum are all divisible (in whole) by the fundamental frequency of the periodic input. For linear dynamical systems, the Pi property always hold true, as the existence of transfer functions dictates that the output frequency grid to be equal to that of the input. In some literature, e.g. Marconato, Van Mulders, Pintelon, Rolain and Schoukens (2010), Pi systems are denoted period-in-same-period-out (Pispo) systems.

In this thesis, the focus is on periodic sequences due to several advantages highlighted in the next section, and as such all nonlinear systems considered are restricted to Pi systems.

2.2 Fourier transformations and periodicity

A signal is a carrier of information and it may take different physical forms: electrical such as voltage values; or mechanical such as the angle of a valve controlling the flow rate of some fluid. A signal can also be either continuous (in amplitude), an analogue signal; or discrete, a digital signal. Throughout this thesis, the term *sequence* is used only to denote digital signals.

The spectral density of an infinitely long aperiodic continuous time signal not described by a finite sum of sinusoids is a continuous function of frequency, and is obtained by the continuous Fourier transform (FT). The Nyquist-Shannon sampling theorem states that the sampling frequency should be more than twice as high as the highest frequency content of a band-limited signal to be able to reconstruct the original signal with no loss of information. When an infinitely long analogue signal is sampled at fixed time intervals to create a sequence, with sampling frequency f_s (or sampling interval $T = 1/f_\mathrm{s}$) chosen to obey the Nyquist-Shannon sampling theorem, the spectral density is given by the discrete-time Fourier transform (DTFT), defined for a sampled input $x[k]$ as:

$$X_{\mathrm{DTFT}}(e^{\mathrm{j}\omega}) = \sum_{k=-\infty}^{\infty} x[k]e^{-\mathrm{j}\omega k} \tag{2.1}$$

where $k = n/f_\mathrm{s}$ for all integer $n \in \mathbb{Z}$. With respect to this equation, the DTFT manifests repeated spectral 'copies' of the FT of the continuous signal, with each copy centred at frequencies $nf_\mathrm{s}\,\forall n \in \mathbb{Z}$. If one normalises the angular frequency variable from $\omega = 2\pi f$, with units of radian per second, to $\omega \longrightarrow 2\pi f/f_\mathrm{s} = 2\pi fT$, the periodicity of the X_{DTFT} over frequency then becomes exactly 1. Provided the Nyquist-Shannon theorem is satisfied, the overlapping or spectral aliasing between different copies is nil. In practice many signals are not truly band-limited and hence there is always a certain amount of aliasing; but with careful measurement strategy, e.g. with proper use of anti-aliasing filter, this can be minimised. In both cases the spectral density is a density function since the distribution is continuous over frequency and has dimension Hz^{-1}.

In reality, it is a necessity to perform measurement in finite time. When a truncation of an infinite signal or sequence is made to create a finite data record,

i.e. applying a rectangular time-window to obtain a snapshot, the record is assumed periodic when transferred to the frequency domain through the discrete Fourier transform (DFT), defined as:

$$X_{\text{DFT}}(k) = \sum_{n=0}^{N-1} x[n] e^{-\mathrm{j}2\pi kn/N} \tag{2.2}$$

for $k = 0, 1, \ldots, N-1$. The principal difference between DFT and FT/DTFT is that the DFT $X_{\text{DFT}}(k)$ is always discrete in frequency. By imposing the input x to have a periodicity of N, $X_{\text{DFT}}(k)$ in essence becomes the N-point sampled version of the X_{DTFT} around the unit circle in the complex plane, at the points $\omega_k = (2\pi k/N)$. This results in a *power spectrum* containing a discrete number of frequency lines, rather than power spectral density. Power spectrum is dimensionless in terms of time (i.e. involves only the unit of measurement, but not Hertz). If the original signal is aperiodic, the DFT is only an approximation due to spectral leakage caused by truncation. Different windowing functions such as the Hanning (named after Austrian meteorologist Julius von Hann) window are traditionally used instead of the rectangular window (simple truncation) to improve the estimate (Blackman & Tukey, 2012). In the case of periodic signals, the Poisson summation formula (developed by French mathematician Siméon Denis Poisson) also shows that the periodic summation of a function in the time domain is described completely by discrete samples of its Fourier transform. The converse is also true—the periodic summation of a function in the frequency domain is described completely by discrete samples of the original function in the time domain (Pinsky, 2002). Hence, in such cases, both the time domain and frequency domain functions are periodic and contain an equivalent description of one another.

In addition to linking a signal or sequence to its power spectrum, FTs are also used for transforming an impulse response function (IRF) (see Section 1.1.3) in the time domain to its frequency response function (FRF) frequency domain counterpart. If the DFT is used, the resulting FRF $G(z)$ is valid for predicting the output of a discrete-time dynamical system $Y(z) = G(z)U(z)$ when subjected to discrete-interval periodic input excitation $U(z)$, in steady-state.

2.3 Periodic input sequences

System identification is concerned with the extraction of dynamics from a system through some form of measurements on the inputs and outputs. In some situations, it is not possible to design an excitation signal or even apply one to a system depend-

ing on its nature. For example, one only has a limited control on the concentration of a drug in a patient. In many problems however, the input signal is only restricted by the maximum amplitude limits of an actuator or by the maximum power a system can handle without its integrity being compromised.

There are two important properties that make an input signal viable for system identification for linear systems. Firstly, a signal must be '*persistently exciting*' (see Shimkin & Feuer, 1987; Söderström & Stoica, 1989, def. 5.1) such that the IRF or FRF can be extracted through mathematical operations; and secondly, a signal should contain maximal spectral content (or power) in the frequency band of interest to maximise the signal-to-noise ratio (SNR) of the estimates, and with enough frequency resolution to discern between features of interest, for example, sharp resonances. The impulse, introduced in Section 1.1.3, is a persistently exciting white signal that contains equal energy content in all frequencies. Impulses however, have a major disadvantage—being a signal that lasts only a very short period of time, the energy transmitted to the system-under-test is limited. When the environment has significant background noise, the impulse response can be overwhelmed and buried by noise, leading to a poor quality measurement. There are many metrics to quantify the suitability of signals for noisy linear system identification, two of which are crest factor (CF) and Performance Index for Perturbation Sequences (PIPS) (see Schoukens, Pintelon, van der Ouderaa & Renneboog, 1988; Godfrey, Barker & Tucker, 1999; Pintelon & Schoukens, 2012, and also Section 5.4). For nonlinear system identification, it can be useful to intentionally suppress spectral lines to gauge the degree of nonlinearity distortions or to minimise them; one example is shown by using the Inverse-repeat Binary Sequence (IRBS) with odd-harmonics suppressed in Section 4.4.

Rather than concentrating the energy in one burst in a short time window, many excitation signals continuously transfer energy into the system, limited only by actuator amplitude or the system energy balance (to avoid overheating, breakdown, etc.). Such signals are now a preference over impulses due to the higher SNR achievable. Thanks to the development of electronics such as digital-to-analogue converters, computers, signal processing algorithms, efficient implementation of Fourier transform algorithms such as the fast Fourier Transform (FFT), more complex signals can be precisely and reliably generated, and measurement results can be interpreted with sufficient computing power.

In this section, several types of sequences mentioned in this thesis are introduced. The use of periodic sequences has an advantage in that the DFT is completely immune to spectral leakage if the sequence was sampled at precisely the right time

points, (i.e. at the correct frequency) to create records of a complete period. Another advantage is that transient effects are easy to minimise or eliminate during measurement. Because of these advantages, periodic sequences should be used as excitations for system identification of Pi systems wherever possible. All of the discrete-time excitations considered in this thesis are periodic. Zero-mean or near zero-mean sequences are used and therefore the dc level (zeroth frequency) is either exactly or approximately zero.

Note that in this thesis, where a periodic discrete-interval sequence is said to have a period of N, the unit is in samples. This gives a period in seconds of NT, where T is the bit-interval of the signal generator in seconds. For the specification of periods hereinafter, the units may be omitted; NT seconds and N samples are used interchangeably.

2.3.1 White Gaussian noise

Aperiodic white Gaussian noise is a typical signal used in modelling environment and measurement noise, in which case the acronym AWGN for additive white Gaussian noise, is often used in the literature. The term 'white' refers to the fact that the power spectral density of the signal has a flat expected power in all frequencies within a certain frequency band of interest—an infinite bandwidth signal does not exist in reality, as such a signal would have infinite energy due to the infinite spectral content. Nevertheless such a mathematical model usually suffices to model a broadband (wide spectrum) noise that is white as far as the band of frequency of interest is concerned. The term *Gaussian* refers to the amplitude distribution or the probability density functions (p.d.f's.) of finding the signal level at any given time as being normally distributed.

Periodic white Gaussian noise may be utilised as excitation signals for system identification. A discrete-interval white Gaussian noise sequence can be obtained when the noise is sampled at fixed time intervals. When the sequence is truncated into multiple records and each of which is made periodic, the magnitudes of their DFT spectra are stochastic. The DFT power spectrum of each record is not strictly white, but after sufficient averaging of many records, the ensemble power spectrum tends to a white spectrum (see Section 4.3).

2.3.2 Discrete-Interval Random Binary Sequences (DIRBS's)

A Discrete-interval Random Binary Sequence (DIRBS) is a discrete-interval sequence of values taking only two levels, a high or a low, with equal probabilities—see, for

example, Godfrey (1980). If the high and low levels are chosen to be $+V$ and $-V$ respectively, the expected value of such a sequence is zero. DIRBS's are aperiodic, and as their length approaches infinity, their autocorrelation functions are asymptotically equal to:

$$R_{\mathrm{uu}}[k] = \begin{cases} V^2 & k = 0 \\ 0 & k \neq 0 \end{cases} \tag{2.3}$$

Through truncation and splitting into records of length N, periodic DIRBS's can be obtained, reaping the benefits associated with periodic sequences in terms of eliminating spectral leakage and allowing for inter-period averaging to reduce the output noise variance. They are not as useful for linear system identification as other sequences because of the need either to use a very large value of N in order for their off-peak autocorrelation function to be small, or equivalently, by using a large number M of different segment realisations with a much smaller value of N and then averaging the results; in either case, a relatively long experimentation time is needed. If M is not large, then it is possible for the DFT at some of the harmonics to be relatively small, resulting in a low SNR at those frequencies—this is discussed in more detail in Section 4.3.

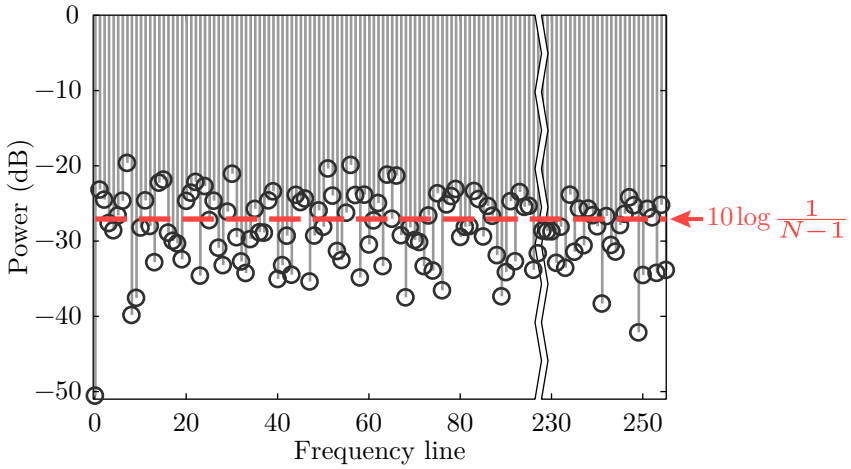


Figure 2.1: DFT power spectrum of a typical periodic DIRBS of period $N = 511$ with levels ± 1

Periodic DIRBS's have similar DFT spectral properties as periodic white Gaussian noise sequences introduced in the previous section. The power spectrum on a decibel (dB) scale of a typical DIRBS of length $N = 511$ is presented in Fig. 2.1. When taken as a whole, the DFT power spectra for a family of zero-mean zero-order hold (ZOH) DIRBS's with power normalised to unity and period N have zero power in dc (zero frequency or the constant term); the remaining power (of unity) is evenly

distributed between the remaining $N-1$ spectral lines. In decibel scale this gives an 'expected value' of power of $10\log_{10}\frac{1}{N-1}$ across the non-zero frequency lines and this is marked in Fig. 2.1 by a dashed line. Note that the y-axis is on a decibel scale, hence the dashed line does not look like a 'line of best fit'. Lastly, because of the stochastic nature, each different realisation of DIRBS has a different power spectrum as seen in the figure; this is in contrast to random-phased multisines (user-defined) and pseudo random sequences, described in later sections.

2.3.3 Random-phased multisines

Periodic multisines are widely used in both linear and nonlinear system identification. Their periodic nature avoids the need to deal with leakage effects. A multisine is a periodic analogue signal constructed by summing a finite number of sine waves with a certain frequency range and phase relationships between them. A multisine $x(t)$ with a linearly spaced frequency grid is defined as:

$$x(t) = \sum_{k=1}^{F} A(k)\sin\big(2\pi k t T_{\mathrm{s}}/T_{\mathrm{N}} + \phi(k)\big) \tag{2.4}$$

where T_{s} is a sampling period of the signal generator, T_{N} is the period of the multisine, F is the maximum number of allowed frequencies, $A(k)$ and $\phi(k)$ are the amplitude and phase for each harmonic k (Schoukens, Pintelon & Rolain, 2012, p. 38). A continuous multisine may be sampled to obtain a digital multisine signal. The user may choose both the power spectrum and the phases of the complex frequency spectrum. A common design choice is to excite a range of frequency band with fixed power and zero power in other frequencies. Appendix B.2 shows the implementation of a MATLAB routine that generates random-phased multisines with flat power spectra.

When the phase of the sine waves at the excited frequencies are designed to follow a continuous uniform distribution such that $\langle\phi\rangle \sim \mathcal{U}(-\pi, \pi)$ (between $-\pi$ and π radians), random-phased multisines are obtained. This is not the only way to design the phases; for example, Schroeder-phased multisines have a special phase structure, resulting in a signal resembling somewhat a chirp[1] (Schroeder, 1970; Boyd, 1986). This thesis focuses on random-phased multisines because of several desirable properties in *nonlinear* system identification: firstly, the Gaussian p.d.f. of the resulting signal in the time domain; secondly, the ability to generate many independent realisations of them with similar properties and thirdly, their

[1] A chirp signal is a sine wave with its frequency component continuously and monotonically increasing (or decreasing) over time

similarity to Gaussian noise in the time domain while having an invariant and user-defined DFT power spectrum. The asymptotic Gaussian p.d.f. property is especially important to the bias-free identification of the Best Linear Approximation (BLA) (see Chapter 3).

A random-phased multisine has a relatively high CF (Schoukens, Pintelon, van der Ouderaa & Renneboog, 1988), compared to, say, a binary sequence. Various literature in the past (see for example, Schroeder, 1970; van der Ouderaa, Schoukens & Renneboog, 1988; Guillaume, Schoukens, Pintelon & Kollár, 1991) has tackled this problem by optimising the phase through various methods to produce a signal with lower CF, meaning that energy content of the resulting signal is more efficiently packed in a tight amplitude profile. Since many actuators are limited by their output amplitude, a high energy content is required for a high SNR. Doing so however inevitably changes the amplitude distribution of the signal, making the resulting multisine signal no longer Gaussian. In this thesis, the Gaussianity of a signal plays an important aspect and hence the CFs are not optimised.

A band-limited multisine may be sampled to give a discrete multisine sequence. In subsequent usage of the term random-phased multisine sequence or simply multisines, the thesis refers to a white discrete Riemann equivalent (see Schoukens, Dobrowiecki, Rolain & Pintelon, 2009; Schoukens, Lataire, Pintelon, Vandersteen & Dobrowiecki, 2009) random-phased multisine with all harmonics up to the Nyquist frequency excited with equal power, essentially creating a white DFT spectrum. The spectral whiteness is necessary for maintaining a fair comparison with the m-sequences detailed in the next section.

2.3.4 Maximum Length Binary Sequences (MLBS's)

Maximum Length Binary Sequences (MLBS's) or m-sequences are discrete-time signals belonging to a class of pseudorandom binary sequences (PRBS's) (Godfrey, 1993; Godfrey, Tan et al., 2005). These are sequences generated with linear autorecursive relations designed to achieve maximum periodicity (Zierler, 1959; Barker, 1967; Golomb, 1981). Some use the term PRBS to refer to MLBS but it is important to note there are other binary sequences that fall under the umbrella of PRBS, for example, quadratic residue binary sequences, Hall binary sequences and twin-prime binary sequences (Godfrey, Tan et al., 2005; Barker, 2008).

The term 'pseudorandom' stems from the fact that these sequences, while having properties of randomness, are actually deterministic. This deterministic nature means that there are a limited number of sequences given any period (or length) N and each one of them is unique. Two of the most important properties

of MLBS's are their perfectly flat DFT power spectra (except dc) and the shift-and-add(multiply) property (see p. 24). The flat power spectra make MLBS's versatile in system identification and the shift-and-add property has implications in nonlinear distortions (see Chapter 4). More properties of MLBS's are listed on p. 23.

Due to their favourable cross-correlation characteristics, deterministic nature and ease of generation through simple shift register circuitry (Golomb, 1981; Norton, 1986/2009; Godfrey, 1993), m-sequences have been used extensively from communications (Simon, Omura, Scholtz & Levitt, 1994) to system identification in a wide range of fields including acoustical (Borish & Angell, 1983), chemical (Amrani, Dowdeswell, Payne & Persaud, 1998), industrial (Godfrey & Moore, 1974; Vilkko & Roinila, 2008) and physiological (Marmarelis, 2004) systems. MLBS's have unique properties when applied to nonlinear system identification, the shift-and-add(multiply) property for example, allows better than normal rejection of nonlinear distortions, specifics of which will be discussed in Section 4.5.1. Another example is that these sequences are especially suitable when estimating higher-order Volterra kernels (see Section 3.4)—normally a time-consuming and inaccurate procedure (Sutter, 1987, 1992; Reed & Hawksford, 1996).

Galois field mathematics

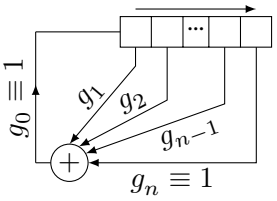


Figure 2.2: Fibonacci implementation of LFSR

MLBS's can be generated from electronic linear feedback shift registers (LFSR) with special feedback topology, which gives the maximum periodic length (period) attainable by traversing all binary $n \geq 3$-bit patterns within the LFSR except one (all zeros) of $N = 2^n - 1$. Any LFSR can be represented by a *generator polynomial* $G(x)$ such that:

$$G(x) = g_0 \oplus g_1 x^1 \oplus g_2 x^2 \oplus ... \oplus g_{n-1} x^{n-1} \oplus g_n x^n \pmod 2 \tag{2.5}$$

where g_n represents the feedback tap configurations and numerically equal to 1 when connected and 0 otherwise. For practical reasons, g_0 and g_n must be set to 1, i.e. connected, see Fig. 2.2. The $G(x)$ is performed in modulo-2 mathematics for binary sequences and obey modulo-2 addition (denoted by $\oplus$) and modulo-2 multiplication

(for the raised powers) operations. Modulo-2 addition is handled electronically by eXclusive-OR logic gates (XORs).

A sequence with maximum period $N = 2^n - 1$ permitted by the number of LFSR bits n is generated only when the feedback topology can be described by a *primitive polynomial*.

Definition $G(x)$ is a *primitive polynomial* if it cannot be factorised into any polynomials of a lesser degree, and it is a factor of the polynomial $x^N + 1 \pmod 2$ where $N = 2^n - 1$, as before, is the period of the MLBS corresponding to the shift register bit length n.

Since the period of the sequence is not a power of 2, the radix-2 fast Fourier transform (FFT) algorithm cannot be used to convert from time domain to the frequency domain, but given the power of modern computers, this is not a significant drawback.

Realisation

An MLBS is obtained by noting the bit symbol of any register over a complete period in a LFSR arrangement. At any given time, the particular bit arrangement in the registers of the LFSR is called a state. An illustrated example in generating a short MLBS ($n = 3, N = 7$) is given below. At this length, the only two valid primitive polynomials are given by $1 \oplus x \oplus x^3$ and $1 \oplus x^2 \oplus x^3$.

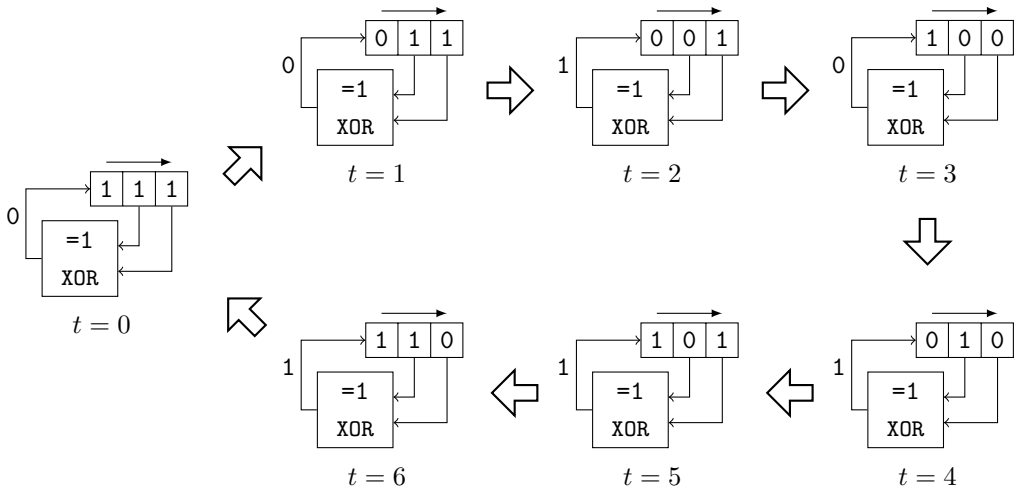


Figure 2.3: Illustrating and the generation of an MLBS of period $N = 7$ using a 3-bit LFSR

Example In Fig. 2.3 there are seven labelled diagrams, each representing a state in time. The electronically high and low levels are denoted '1' and '0' respectively. The 3-bit LFSR takes a single input from the left hand side and shifts the symbols within each cell from left to right every time step in each of the diagrams. The XOR gate takes two feedback taps as input and the output of which becomes the input to the LFSR at the next time sample. The valid feedback taps that generate an MLBS are predetermined as mentioned before and in this example the primitive polynomial $1 \oplus x^2 \oplus x^3$ is used (apply (2.5) whilst comparing Fig. 2.2 and Fig. 2.3).

Table 2.1: XOR gate truth table

Inputs		Output
A	B	
0	0	0
0	1	1
1	0	1
1	1	0

The outputs of the XOR gates according to Table 2.1 are annotated next to the connection between the XOR outputs and the LFSR inputs in Fig. 2.3. The LFSR is initialised arbitrarily at $t = 0$ to '111'. The XOR operation on the second and third register values outputs a single '0' and this then becomes the first register value at $t = 1$. The operation continues in a similar manner until $t = 6$ when the next state change results in an identical state to that of $t = 0$. The MLBS is obtained by reading the value of any register throughout one period of operation. For this example, the MLBS generated is $\langle 1110010 \rangle$. Shifting the sequence in a cyclic manner simply adds phase to the DFT spectrum and does not affect its power spectral properties. Note that there is a forbidden state with shift registers holding '000'. This trivial state is stable but is never reached when generating an MLBS. This explains why the period of an m-sequence generated by an n-bit LFSR is given by $2^n - 1$ rather than 2^n.

Conversion to a sequence with levels $\pm V$ is made by substituting $1 \to +V$ and $0 \to -V$ or by $1 \to -V$ and $0 \to +V$. Their mean level is $\pm V/N$ and their autocorrelation function is given later in this section.

Number of available independent MLBS's

Given a family of MLBS's with a common period of N, there is only a limited number of MLBS's. Refer to the table in Appendix A.1 for the total number of available MLBS's against the n number of bits of the LFSR and the resulting period N of the MLBS. Tables of LFSR feedback topologies are also available for n up to about 30 (New Wave Instruments, 2010). The number of available MLBS's for a given period is always an even number; this is due to the fact that an MLBS reversed in time, called its mirror counterpart, is itself another MLBS and as such, there is always an even number of them given any periodicity. In the example above for $n = 3$ where

there are only two valid feedback tap choices, the two are mirror counterparts of each other. There are several different MLBS for any value of $n > 3$. However, for small n, the number of independent m-sequences is comparatively limited—for example, for $n = 7 \Leftrightarrow N = 127$, there are 18 independent MLBS's, while for $n = 13 \Leftrightarrow N = 8191$, there are 630 including their time-reversed mirror counterparts.

Note that the MLBS's in a given family of common periodicity are assumed to be statistically *independent* of each other, such that the occurrence of a high or low level of a sequence at any one time does not affect its probability on another sequence from the same family, which includes the mirror counterparts.

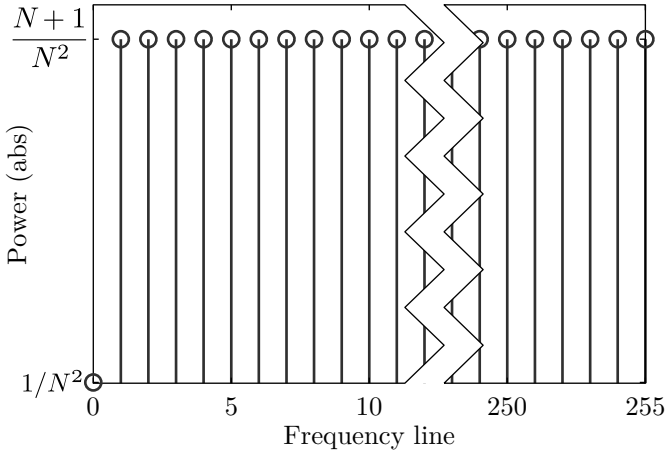


Figure 2.4: DFT power spectrum of a typical MLBS of period $N = 511$ with levels ± 1

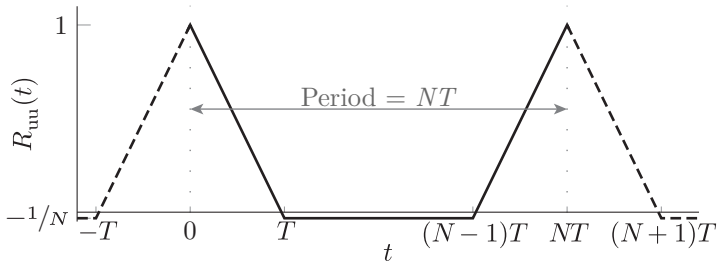


Figure 2.5: Periodic autocorrelation function of a ZOH MLBS with bit-interval of T, period NT and levels ± 1

Properties of MLBS's

Some properties of MLBS's include (Mitra, 2008):

1. *Register states:* An MLBS is generated by traversing all possible states of LFSR except one where all registers are zero. The latter is a stable but trivial state that is never reached, giving the period of MLBS to be equal to the number of all possible register states (2^n) minus the trivial state resulting in $2^n - 1$ where n is the number of registers in the LFSR.

2. *Power spectrum:* All MLBS's have perfectly white (flat) DFT power spectrum for all frequencies except dc (zero frequency), see Fig. 2.4 for the power spectrum of a typical MLBS with power normalised to unity (signal levels ± 1).

3. *Mirror counterparts:* A time-reversed copy of a period of MLBS is itself another MLBS, called the mirror counterpart. The mirror counterpart of any MLBS can also be generated by reversing the feedback tap configuration of the LFSR.

4. *Autocorrelation properties:* The autocorrelation function $R_{\text{uu}}(t)$ of an MLBS approximates an impulse, or a normalised Dirac delta function. With the power normalised to unity (signal levels ± 1), numerically:

$$R_{\text{uu}}(t) = \begin{cases} 1 & t = 0 \\ 1 - \dfrac{(N+1)\,t}{NT} & |t| < T \\ -1/N & T \leq |t| \leq (N-1)T \end{cases} \tag{2.6}$$

   where T is the time interval a bit symbol of the ZOH MLBS signal, and R_{uu} has the period of NT, i.e. same as the signal. This is plotted in Fig. 2.5, from which one can see between the time of $-T$ and T the autocorrelation function is a triangle. The small non-zero off-peak values are an intrinsic property of MLBS's. As N tends to infinity the off-peak values become negligible and $R_{\text{uu}}(t)$ tends to a true Dirac delta. Operating in the discrete-time domain the autocorrelation function $R_{\text{uu}}[k]$ approximates a Kronecker delta function—a discrete-time analogue of the Dirac delta function (Hartmann, 1997, pp. 152–153).

5. *Running length of symbols:* The maximum running length of identical symbols is always equal to the number of bits of the shift registers used to generate the MLBS. For MLBS of period $N = 2^n - 1$ and n being the number of shift register bits, there is always precisely one longest run of '1's of length k equal to n, one run of length $k = n-1$ of '0's and equally one run of length $k = n-2$ for both symbols of '1' and '0'. There are two, four, eight (in increasing powers

of two) for each decreasing length of $k \leq n-3$ for both symbols until $k = 1$, where there are equally 2^{n-3} isolated symbols of '1' and of '0'. Referring back to the LFSR setup in Fig. 2.3, the nature of XOR logic gates means the longest sequence of high must be sandwiched by two low levels at either side, guaranteeing there are no symbols runs longer than n.

6. *Shift-and-add(/-multiply) property:* The modulo-2 sum of an MLBS $\langle u \rangle$ with levels '1' and '0' and a time-delayed version of itself $\langle u[t-a] \rangle$ results in another time-delayed version of itself $\langle u[t-b] \rangle$ (Zierler, 1959). The specific delay relation between the original and the resulting sequence is a function of the specific MLBS $\langle u \rangle$ and the value of a, such that $b = \mathfrak{f}(a, \langle u \rangle)$. This property is called the *shift-and-add property* and is unique to m-sequences. If instead the sequence is symmetric (with signal levels $\pm$a), the property is then called the *shift-and-multiply* property.

Generating MLBS with computer software packages

MLBS's can be generated using the MATLAB software and the third-party tool *prs* (Tan & Godfrey, 2002) or the stand-alone program GALOIS (Barker, 2008; Godfrey, Tan et al., 2005). The MATLAB program code utilising the *prs* tool is included in Appendix B.1.

2.3.5 Inverse-Repeat (Maximum Length) Binary Sequences (IRBS's)

An IRBS u_{IRBS} is an MLBS u_{MLBS} with every other symbol or sample inverted, creating a sequence twice the period of the original MLBS, such that:

$$\begin{cases} u_{\text{IRBS}}[k] = +u_{\text{MLBS}}[k], & k = 1, 3, 5..., 2N-1 \\ u_{\text{IRBS}}[k] = -u_{\text{MLBS}}[k], & k = 2, 4, 6..., 2N \end{cases} \tag{2.7}$$

where $N = 2^n - 1$ is the period of the parent MLBS and $u_{\text{MLBS}}[k + iN] \equiv u_{\text{MLBS}}[k]$ for all integers i and k due to the periodicity.

IRBS's have a white DFT power spectrum, though the spectrum has only power in the odd harmonics. Also, the dc frequency has exactly zero power, which means the sequences have exactly zero mean in the time domain. Fig. 2.6 shows a typical power spectrum of an IRBS with levels ± 1 generated by an MLBS of period $2^8 - 1 = 255$, resulting in the IRBS with a period of $255 \times 2 = 510$.

For nonlinear system identification, the lack of power at even harmonics is a highly desirable property, because it removes the effects of even order system nonlinearities (Schoukens, Swevers, Pintelon & van der Auweraer, 2004). One can

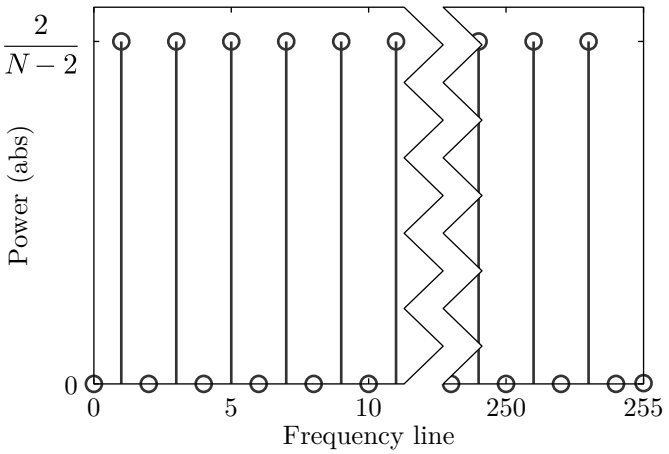


Figure 2.6: DFT power spectrum of a typical IRBS of period $N = 510$ with levels ± 1

observe this also equates to the frequency resolution being halved compared to the related MLBS. To preserve the frequency resolution, it is necessary to halve the sampling time, which is not always feasible; or doubling the experiment time, which is undesirable.

2.3.6 Multilevel sequences

Multilevel sequences in this thesis refer to discrete-level signals with more than two signal levels, for example, ternary (3-level), quaternary (4-level), quinary (5-level) and so forth. These sequences may be truly random or pseudorandom (see previous section), in the latter case they are called pseudorandom multilevel sequences (PRMS's). In addition, there are no restrictions placed on the actual discrete distribution of the signal levels; it can be symmetric (with equal positive and negative distribution of signal levels), or asymmetric and arbitrary. The discrete distribution of signal levels can be defined by the *probability mass function* (p.m.f.), a discrete analogue of the continuous and more commonly known *probability density function* (p.d.f.).

2.4 Conclusions

This chapter introduced several types of input sequences known in the literature, which are discussed and used in the experiments throughout this thesis. This sets the stage for Chapter 3, where theoretical expressions for the Best Linear Approximation (BLA) are derived, applicable to all input sequences introduced in this chapter.

Chapter 3

The Best Linear Approximation

Nonlinear systems are systems where the input-output relationships do not obey the principle of superposition. An example of such a system is the human auditory system—doubling the power output of a loudspeaker does not double the perceived loudness. All systems are nonlinear to some extent and the term 'linear system' is used to describe systems that are predominantly linear. For example, an electrical circuit with only capacitors, inductors and resistors and a driving electromotive force, or the mechanical analogue of mass, spring, damper and a driving force; are described as linear systems. However, under extreme situations such as when the driving voltages or forces are too high, the resistors may overheat, the spring may be over-compressed; the linear relationships may no longer apply. The opposite is also true, small perturbations in a well-behaved nonlinear system can be predominantly linear around a small region, called the operating point. Linearising a nonlinear system around such a point has merits in modelling and control.

In linear system identification, both the frequency response function (Frf) and the impulse response function (Irf) of a system contain a complete description of the system dynamics, from which it is possible to model, make predictions and control the system to achieve desired behaviour. For nonlinear systems, a complete description is infeasible to obtain, unless one has a complete physical knowledge of the system. Therefore, one is better off modelling the system across a certain operating range. One of the most familiar ways of linearising a nonlinear system about an operating point is to use the Best Linear Approximation (Bla) (Enqvist & Ljung, 2005; Pintelon & Schoukens, 2012; Mäkilä & Partington, 2004; Mäkilä, 2004, 2006). The Bla depends upon the amplitude distribution of the excitation (e.g., binary, uniform, Gaussian) and on the power spectrum of the excitation. In

a standard situation, Gaussian distributed excitations with a user-defined power spectrum are used (Pintelon & Schoukens, 2012; Schoukens, Pintelon, Dobrowiecki & Rolain, 2005), with the model errors minimised in mean square sense (Enqvist & Ljung, 2005; Enqvist, 2005). The major motivation for this choice is that, under these conditions, the theoretical insights are very well developed (Pintelon & Schoukens, 2012; Schoukens, Lataire et al., 2009). However, Gaussian excitations are definitely not always the simplest or the best choice from a practical point of view.

In this chapter, the effect of the distribution of the excitation signal on the linear approximation is studied. For example, what happens if a binary excitation is used instead of a Gaussian noise excitation? Will the results significantly change, or will the differences be sufficiently small that, for practical purposes, they can be ignored? Is it possible to quantify the differences between non-Gaussian and Gaussian excitations on the basis of the non-Gaussian measurement results? These are important questions for the measurement community, because in some applications, for example, opening and closing a valve, it may be very difficult or even impossible to apply a Gaussian excitation, while it could be very easy to use a binary excitation.

3.1 Introduction to the BLA

Definition The Best Linear Approximation (BLA), when applied to a time invariant and periodicity-invariant (see Section 2.1) nonlinear system, is a specific impulse response function (in the time domain) or transfer function (in the frequency domain) that minimises the squared differences between the output computed from a linear dynamic filter with transfer characteristics of the BLA to the output of the nonlinear system in question.

Mathematically, this may be expressed as:

$$g_{\text{BLA}} = \arg\min_{g} \mathrm{E}\left[\!\left[|y - g * u|^2 \right]\!\right] \tag{3.1}$$

where u is the input to the system, y is the corresponding measured system output, the tuning parameter g is a linear IRF, and E is the expectation operator which operates across all input u from a common statistical process $\mathcal{X}$ such that $\langle u \rangle \sim \mathcal{X}$. The symbol $*$ denotes convolution such that $g * h(t) \triangleq \int_{-\infty}^{\infty} g(\tau)h(t-\tau)\mathrm{d}\tau$ for continuous functions and $g * h[n] \triangleq \sum_{m=-\infty}^{\infty} g[m]h[n-m]$ for discrete functions.

An equivalent relationship in the frequency domain for (3.1) may also be established as:

$$G_{\mathrm{BLA}} = \arg\min_{g} \mathrm{E}\left[\!\left[(Y - GU)\overline{(Y - GU)}\right]\!\right]. \tag{3.2}$$

Eykhoff (1974), Bendat and Piersol (1980) have shown that, for *any* nonlinear system with an input signal u and with output signal y, an estimator of g_{BLA} can be constructed as follows: The least squared errors between the nonlinear output modelled by a linear BLA filter are minimised (Enqvist & Ljung, 2005; Enqvist, 2005) such that:

$$y \approx g_{\mathrm{BLA}} * u.$$

Moving into the frequency domain and assuming input U to be a white signal U' filtered by a filter H such that $U = U'H$, then:

$$Y \approx G_{\mathrm{BLA}} \cdot U'H.$$

Cross-correlating both sides with the input U gives:

$$\begin{aligned} Y \cdot \overline{U'H} &\approx G_{\mathrm{BLA}} \cdot U'H \cdot \overline{U'H} \\ Y \cdot \overline{U'} \cdot \overline{H} &\approx G_{\mathrm{BLA}} \cdot H\,U'\overline{U'}\,\overline{H} \\ G_{\mathrm{BLA}}(\mathrm{j}\omega) &\approx \frac{Y\overline{U}}{H\,U'\overline{U'}\,\overline{H}} = \frac{Y\overline{U}(\mathrm{j}\omega)}{R(\omega)}. \end{aligned} \tag{3.3}$$

For a periodic input signal of period N and sampling time T, (3.3) is modified to:

$$G_{\mathrm{BLA}}(\mathrm{j}\omega_k) \approx \frac{Y\overline{U}(\mathrm{j}\omega_k)}{R(\omega_k)}. \tag{3.4}$$

Here $R(\omega_k)$ is the (auto-)power spectrum of one specific realisation of input u, $Y\overline{U}(\mathrm{j}\omega_k)$ is the cross-power spectrum between u and y and $\omega_k = 2\pi k/NT$ where $k \in \{0, 1, \ldots, N-1\}$ is the harmonic number. Given a single measurement record of the input u and output y, (3.3) gives one specific estimate. In practice, a good BLA estimate requires an aggregate asymptotic convergence of many independent experiments with different input realisations, each generated by a common parent process. With any single experiment realisation, there are inevitably deviations from the BLA, hence the approximately equals sign in (3.3). In a scenario *devoid* of exogenous environment and measurement noise, any deviation of the actual output y of a time-invariant nonlinear system from the expected output due to linear contributions $y_{\mathrm{L}} = g_{\mathrm{BLA}} * u$ for a given realisation of input u may be classified as

nonlinear distortions. One can write:

$$\begin{aligned} y &= y_{\mathrm{L}} + \nu \\ &= g_{\mathrm{BLA}} * u + \nu \end{aligned} \tag{3.5}$$

where ν, the nonlinear distortion term, is a function of system input and the structure of the nonlinearity. Given a class of periodic input $\langle u \rangle$, this nonlinear distortion term will also be periodic and is specific to a particular realisation of u—this is in contrast to exogenous environment noise, which is usually aperiodic in nature. This has implications in the averaging strategy required to obtain good estimates of BLA in a non-parametric identification problem—one has to utilise different realisations of inputs, rather than periodic renditions of a single realisation, to drive down nonlinear distortions and therefore uncertainty of the BLA estimates.

With this in mind, the usual method of estimating the BLA in the frequency domain, is to introduce averaging to (3.3) to give:

$$G_{\mathrm{BLA}} = \frac{\mathrm{E}[\![Y\overline{U}]\!]}{\mathrm{E}[\![R]\!]} \tag{3.6}$$

where E is the expectation operator, numerically performed through averaging schemes such as the arithmetic mean. The expectation operator operates across different realisations of u. For white input this is then:

$$G_{\mathrm{BLA}} = \frac{\mathrm{E}[\![Y\overline{U}]\!]}{\mathrm{E}[\![U\overline{U}]\!]} \tag{3.7}$$

$$= c\mathrm{E}[\![Y\overline{U}]\!] \tag{3.8}$$

where c is a constant; as the 'population' auto-power spectrum $\mathrm{E}[\![U\overline{U}]\!]$ is a constant by definition of spectral whiteness. In the time domain for white inputs $\langle u \rangle$, this is equivalent to:

$$g_{\mathrm{BLA}} = c\,\mathrm{E}[\![u \star y]\!]$$

where $\star$ is the *cross-correlation* operator such that $g \star h(t) \triangleq \int_{-\infty}^{\infty} g(\tau)h(t+\tau)\mathrm{d}\tau$ for continuous functions and $g \star h[n] \triangleq \sum_{m=-\infty}^{\infty} g[m]h[n+m]$ for discrete functions. The constant c is unity if the power of u is unity. Introducing and incorporating the time variable of interest r one obtains:

$$g_{\mathrm{BLA}}(r) = c\,\mathrm{E}[\![u(t)y(t+r)]\!].$$

To make causality easier to deal with, negative shift variables (to obtain past samples) are preferable. Use the fact that $(m \star n)(t) \equiv (n \star m)(-t)$ to obtain:

$$\begin{aligned} g_{\text{BLA}}(-r) &= c\,\text{E}[\![y(t)u(t+r)]\!] \\ g_{\text{BLA}}(r) &= c\,\text{E}[\![y(t)u(t-r)]\!]. \end{aligned} \tag{3.9}$$

This equation will form a basis for deriving the theoretical BLA in Sections 3.3 and 3.4.

It is important to realize that $g_{\text{BLA}}(r)$ depends not only on the first-order kernel in the Volterra expansion of the nonlinear system, but also on the higher order kernels (Schetzen, 1980/2006). For this reason, the BLA depends on the higher order moments of the input which, in turn, makes it dependent on the distribution of the input signal. Even order nonlinearities do not contribute to the BLA for input distributions that are symmetric about zero. This result is also in agreement with the early result of Bussgang (1952) which states that, for coloured Gaussian noise applied to a static nonlinearity, the following relation holds between the input-output cross-correlation function $R_{\text{yu}}(\tau)$ and the input autocorrelation function $R_{\text{uu}}(\tau)$:

$$R_{\text{yu}}(\tau) = c \cdot R_{\text{uu}}(\tau) \tag{3.10}$$

where c is a constant that depends only on the nonlinear amplitude characteristic.

In Section 3.2, the concept of higher order moments for a signal or sequence is defined. The theoretical BLA is first developed for a more basic block structured nonlinear system, namely the Wiener-Hammerstein (WH) in Section 3.3; then this is extended to the more general Volterra system in Section 3.4. In both cases, let the system be excited by M random realisations of independent and identically distributed sequences $u_m[k]$, $1 \leq m \leq M$, all of period N; the following assumptions are made:

1. Steady state is reached before the system output is measured.
2. Input and output measurements are synchronised so that there are no leakage effects.
3. M is sufficiently large so that the deviation of the measured output frequency spectrum (averaged over the M realisations) from the true output spectrum of the linear element can be regarded as due to nonlinear distortions but not disturbance noise.

3.2 Moments

Definition The n^{th} moment of a periodic random sequence u with notation $\mathfrak{M}_n$ is defined as:

$$\mathfrak{M}_n \triangleq \mathrm{E}[\![u^n]\!] \tag{3.11}$$

where E is the expectation operator acting across all possible sequences of the same random process from which the sequences were generated.

The first moment when $n = 1$ is the *mean* while the second moment when $n = 2$ is equal to the *variance* when the mean is zero. The second moment also relates to the power of the signal or sequence. The term 'higher order moment' is used to describe moments where $n \geq 3$ and is studied in a branch of statistics called higher order statistics.

Table 3.1: Moment values for zero-mean Gaussian, uniform binary and uniform ternary sequences

Degree	$\mathfrak{M}_n$		
n	Gaussian	Binary	Ternary
1	0	0	0
2	σ^2	a^2	$2b^2/3$
3	0	0	0
4	$3\sigma^4$	a^4	$2b^4/3$
5	0	0	0
6	$15\sigma^6$	a^6	$2b^6/3$
7	0	0	0
8	$105\sigma^8$	a^8	$2b^8/3$
⋮	⋮	⋮	⋮
n even	$(n-1)!!\,\sigma^n$	a^n	$2b^n/3$

The *central* moment is the moment of random sequences with mean equal to zero, and these can be derived for Gaussian noise, symmetrical uniform binary sequences including Discrete-interval Random Binary Sequences (DIRBS's) and Maximum Length Binary Sequences (MLBS's) with levels $\pm a$ and symmetrical uniform ternary sequences with levels 0 and $\pm b$. For these three types of sequences, the central moments are tabulated in Table 3.1. The term 'uniform' means that the sequence levels have equal probability of appearance, and the term 'symmetrical' refers to the symmetry of the amplitude distribution about the mean. With mean equal to zero, all odd degrees moment terms are zero. The values of even order central moments for Gaussian, uniform binary and ternary sequences are tabulated

in Table 3.1. For MLBS's with a period of N, the mean $= \pm a/N$ is only approximately zero but can be regarded as so in practice since N is usually chosen to be large (> 127). Hence, in practice, the uniform binary result can be considered to be valid for MLBS's for large N.

In the table, the operator !! represents the double factorial operator, which is defined for any positive odd integer x as $x!! \triangleq x(x-2)(x-4)\ldots 1$. Note that if the power of these sequences is normalised to unity, $\mathfrak{M}_2 = 1$ and hence $\sigma^2 = 1$, $a^2 = 1$ and $2b^2/3 = 1 \Rightarrow b^2 = 1.5$.

An important aspect is the ergodic convergence properties for MLBS's—it is not obvious that the expected higher order moments of MLBS's over the finite set of realisations for a given period N are equal to those of binary noise (i.e. DIRBS's). However, the extensive simulations performed in Section 4.4 suggest that the initial convergence over the number of realisations is faster for the MLBS's than for the DIRBS's (see Fig. 4.5 and Fig. 4.6).

3.3 Discrete-time Wiener-Hammerstein systems

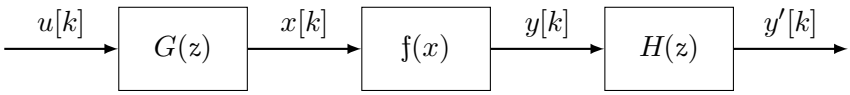


Figure 3.1: Wiener-Hammerstein system structure

Fig. 3.1 shows the block structure of a single-input single-output (SISO) time-invariant Wiener-Hammerstein (WH) nonlinear system with a static nonlinearity. The WH system is one of the simplest block structured nonlinear system and falls under the umbrella of Volterra systems. It is a slightly more general form of the yet simpler Wiener (without the second linear filter, i.e. $H(z) = 1$) and Hammerstein (without the first linear filter, i.e. $G(z) = 1$) systems. For input signals $\langle u \rangle$ with a Gaussian amplitude distribution such that $u \sim \mathcal{N}(\mu, \sigma^2)$, Dobrowiecki and Schoukens (2002), Schoukens, Lataire et al. (2009) & Pintelon and Schoukens (2012) have shown that the BLA of a WH system is given by:

$$\hat{G}_{\text{BLA, Gaussian}} = c\,G(z)H(z) + \mathcal{O}(F^{-1}), \tag{3.12}$$

where F is a bias term dependent upon the number of excited frequencies of the random-phased multisines used in the identification. Here, c is a constant that depends on the higher order Volterra kernels and the power spectrum (root mean

squared (rms) value and colouring) of the input signal (Schoukens, Lataire et al., 2009; Pintelon & Schoukens, 2012).

If the perturbation signal is not Gaussian-like, then (3.12) does not hold, and it is the purpose of this section to determine how (3.12) should be modified for some commonly used non-Gaussian perturbation signals. For the WH system structure shown in Fig. 3.1, there are several important assumptions:

1. The overall WH system is periodicity-invariant (Pi), causal, stable and time-invariant.
2. $G(z)$ and $H(z)$ are stable, causal and time-invariant filters.
3. The input u is spectrally white, stationary, known, has zero mean and persistently exciting (see Shimkin & Feuer, 1987; Söderström & Stoica, 1989, def. 5.1). The input signals introduced in Section 2.3 are all stationary and persistent excitations. While the expressions derived are valid only for zero-mean signals, in practice having inputs with non-zero mean would only upset the operating point and introduce a dc bias in the estimated BLA—a dc bias can be accounted for easily.
4. The intermediate output y also needs to be persistently exciting. To visualise why, consider a pure Hammerstein system such that $G(z) = 1$; if u is binary and the nonlinearity only contains even order polynomial degrees (e.g. $\mathfrak{f}(x) = x^2$), y will be a constant and therefore not persistently exciting. In such a case, it is then not possible to extract the dynamics of $H(z)$.

As long as these assumptions are satisfied, the theory of the BLA in this section applies to both Hammerstein and Wiener systems; as they are degenerate cases of WH systems.

The analysis in the remainder of this section can be simplified somewhat by noting that, in the right-hand part of (3.12), the term $cG(z)$ is followed by the transfer function $H(z)$, which, being linear by definition, has a linear effect on the output $y[k]$ so that the overall system can be regarded as a Wiener system followed by a linear system. The corresponding equation for a perturbation signal that is not Gaussian-like is given by:

$$\hat{G}_{\text{BLA, non-Gaussian}} = c'\,G'(z)H(z) + V(z) \tag{3.13}$$

where $V(z)$ is a term accounting for the nonlinear distortions in the frequency domain.

Since the purpose of this section is to see how c' and $G'(z)$ in (3.13) differ from c and $G(z)$ in (3.12), without loss of generality it is possible to set $H(z) = 1$ and to use the Wiener system shown in Fig. 3.2 instead.

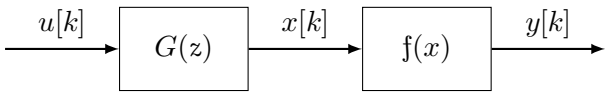


Figure 3.2: Wiener system structure

Consider a Wiener system with the block structure in Fig. 3.2, with the linearity being a causal and stable finite impulse response (FIR) filter of an arbitrary order. Note that the theory developed is also applicable to stable infinite impulse response (IIR) filters, because an IIR filter can be regarded as a FIR filter of an infinite degree. The Pi property described in Section 2.1 applies—if the input u is periodic, the output y has the same periodicity as u. Assuming the static nonlinearity may be represented with reasonable accuracy by a finite polynomial such that $\mathfrak{f}(x) = p_0 + \sum_{i=1}^{q} p_i x^i$, and that the input u is periodic, then a period of the output y is given by:

$$y[t] = \sum_{i=1}^{q} p_i \big(g \circledast u[t]\big)^i + p_0 \tag{3.14}$$

where $\circledast$ is the circular convolution operator such that for $0 \leq t < N$:

$$g \circledast u[t] \triangleq \sum_{k=0}^{N-1} g[k]u[t-k], \tag{3.15}$$

t is an integer, N is the periodicity and p_0 is a constant. Through multinomial expansion:

$$\begin{aligned}\left(\sum_{k=0}^{N-1} x_k\right)^i &\equiv \sum_{k_1=0}^{N-1}\sum_{k_2=0}^{N-1}\cdots\sum_{k_i=0}^{N-1} x_{k_1}x_{k_2}\cdots x_{k_i} \\ &= \sum_{k_1=0}^{N-1}\cdots\sum_{k_i=0}^{N-1}\prod_{\alpha=1}^{i} x_{k_\alpha},\end{aligned} \tag{3.16}$$

substituting (3.15) into (3.14) and expanding gives:

$$y[t] = p_0 + \sum_{i=1}^{q} p_i \sum_{k_1=0}^{N-1}\cdots\sum_{k_i=0}^{N-1}\prod_{\alpha=1}^{i} g[k_\alpha]u[t-k_\alpha]. \tag{3.17}$$

Finally, using (3.9) and ignoring the constant term p_0 of the polynomial nonlinearity, the BLA of a discrete-time Wiener system with linearity IRF of g and a static polynomial nonlinearity up to degree q with coefficients p_i, subjected to input $\langle u \rangle$ is

given by:

$$g_{\text{BLA}}[r] = \mathrm{E}\left[\!\!\left[\sum_{i=1}^{q} p_i \sum_{k_1=0}^{N-1} \cdots \sum_{k_i=0}^{N-1} \prod_{\alpha=1}^{i} g[k_\alpha]u[t-k_\alpha]u[t-r]\right]\!\!\right]. \tag{3.18}$$

The theory here puts a focus on polynomial static nonlinearities for the Wiener structure described; the reason for this is that any causal, finite-memory and finite-dimensional discrete-time Volterra series can be exactly represented by a finite sum of parallel Wiener systems, and the use of polynomial nonlinearities results in rapid convergence (Korenberg, 1991). Since the input-output relationship in a parallel structure obeys the principle of superposition, the theory for the BLA developed in this chapter is applicable to a wide range of systems. The theory for a general Volterra system will be discussed in Section 3.4.

3.3.1 Pure cubic nonlinearity

Consider a pure cubic nonlinearity so that $\mathfrak{f} \mapsto p_3 x^3$ and $q = 3$ in (3.18), its BLA is given by:

$$\begin{aligned} g_{\text{BLA}}[r] &= \mathrm{E}\left[\!\!\left[p_3 \sum_{k_1=0}^{\infty} \cdots \sum_{k_3=0}^{\infty} \prod_{\alpha=1}^{3} g[k_\alpha]u[t-k_\alpha]u[t-r]\right]\!\!\right] \\ &= p_3 \sum_{k_1=0}^{\infty}\sum_{k_2=0}^{\infty}\sum_{k_3=0}^{\infty} g[k_1]g[k_2]g[k_3]\mathrm{E}\left[\!\!\left[u[t-k_1]u[t-k_2]u[t-k_3]u[t-r]\right]\!\!\right]. \end{aligned} \tag{3.19}$$

Without loss of generality choose the linearisation operating point such that the input may be treated as a zero-mean signal, so that all odd higher moment terms are effectively zero (see Table 3.1 of Section 3.2). Within the summation, only two patterns result in non-trivial (non-zero) contribution to the BLA; this is when the combination result in even order moment terms in the expectation operator. The two patterns are:

$$\begin{cases} g^2[k]g[r]\mathrm{E}\left[\!\!\left[u^2[t-k]u^2[t-r]\right]\!\!\right] & \text{for } k = k_1 = k_2 \neq r, k_3 = r \\ g^3[r]\mathrm{E}\left[\!\!\left[u^4[t-r]\right]\!\!\right] & \text{for } k = k_1 = k_2 = k_3 = r. \end{cases} \tag{3.20}$$

Using the moment notation in Section 3.2 simplifies the notation to:

$$\begin{cases} g^2[k]g[r]\,\mathfrak{M}_2^2 & \text{for } k = k_1 = k_2 \neq r, k_3 = r \\ g^3[r]\,\mathfrak{M}_4 & \text{for } k = k_1 = k_2 = k_3 = r. \end{cases} \tag{3.21}$$

Note that in the first pattern the choice of k_1, k_2 and k_3 is arbitrary, giving the binomial coefficient $\binom{3}{1} = 3$ degrees of freedom due to permutation. Expressing the patterns in the sum gives:

$$g_{\text{BLA}}[r] = \mathfrak{M}_4\, p_3 g^3[r] + 3\,\mathfrak{M}_2^2\, p_3 g[r] \sum_{\substack{k=0 \\ k\neq r}}^{\infty} g^2[k]. \tag{3.22}$$

Gaussian input case

Now consider the input having a Gaussian amplitude distribution, such that the moment terms $\mathfrak{M}_2 = \sigma^2$ and $\mathfrak{M}_4 = 3\sigma^4$ as noted in Table 3.1, (3.22) becomes:

$$g_{\mathcal{N}\,\text{BLA}}[r] = 3p_3\sigma^4 g^3[r] + 3p_3\sigma^4 g[r] \sum_{\substack{k=0 \\ k\neq r}}^{\infty} g^2[k].$$

Note the restriction that $k \neq r$ in the summation term. To remove this restriction, consider $g^3[r] = g[r]g^2[r]$ hence the second term may be incorporated back into the summation to form:

$$g_{\mathcal{N}\,\text{BLA}}[r] = 3p_3\sigma^4 g[r] \sum_{k=0}^{\infty} g^2[k]. \tag{3.23}$$

Binary input case

On the other hand, if the input is binary, such that the moment terms $\mathfrak{M}_2 = a^2$ and $\mathfrak{M}_4 = a^4$ as noted in Table 3.1, (3.22) becomes:

$$g_{\mathcal{U}_2\,\text{BLA}}[r] = p_3 a^4 g^3[r] + 3p_3 a^4 g[r] \sum_{\substack{k=0 \\ k\neq r}}^{\infty} g^2[k].$$

This time, unlike the Gaussian case, there is a coefficient mismatch. For the second term to be incorporated back into the summation, a compensation factor ϵ is required:

$$g_{\mathcal{U}_2\,\text{BLA}}[r] = \epsilon p_3 a^4 g^3[r] + 3p_3 a^4 g[r] \sum_{k=0}^{\infty} g^2[k] \tag{3.24}$$

where $\epsilon = 1-3 = -2$. With the power (second order moments $\mathfrak{M}_2$) of both Gaussian and binary inputs normalised to unity, one can write the binary BLA in terms of the Gaussian BLA as:

$$g_{\mathcal{U}_2\,\text{BLA}}[r] = g_{\mathcal{N}\,\text{BLA}}[r] - 2p_3 g^3[r]. \tag{3.25}$$

Similar expressions may be obtained through the elementary algebraic polynomial expansion of the system output y (Wong et al., 2012a).

Note that for MLBS inputs, (3.24) is still valid with one caveat—the IRF of the BLA would have a small constant bias due to MLBS's having non-zero mean of $\pm a/N$ where a is the amplitude and N is the period (see Section 1). This small bias is visible with $N = 127$ in Fig. 4.7a, where short negative blips can be seen below the x-axis for $r \geq 3$. This is not a problem as N is usually chosen to be larger than 127. Even if that is not possible, a dc bias in the BLA can be easily accounted for. Lastly, while the expression for the BLA is asymptotic with respect to nonlinear distortions, in practice the deterministic power spectra of an MLBS (see Section 4.3) and behaviour of nonlinear distortions (introduced in Section 4.5) allow MLBS's to achieve a convergence quicker than DIRBS's. As such, MLBS's require fewer realisations to reach a given level of variance of the BLA estimate. This also means the finite number of MLBS for a given period (see Appendix A.1) is, in most cases, not a concern.

Arbitrary input case

For a cubic Wiener system, one can observe that the discrepancy between the BLA obtained from Gaussian sequence and that obtained from binary sequence depends both on the linearity IRF and the nonlinearity. For a signal with an arbitrary amplitude distribution, following similar procedures one can write the BLA obtained from an arbitrary sequence, with power normalised to unity, in terms of the Gaussian BLA as:

$$g_{\mathcal{X}\,\mathrm{BLA}}[r] = g_{\mathcal{N}\,\mathrm{BLA}}[r] + \delta_4 p_3 g^3[r]. \tag{3.26}$$

Here, the non-Gaussian moment correction terms δ_n are the numerical difference between Gaussian higher order moment of order n with that of an arbitrary sequence, such that:

$$\delta_n \triangleq \mathrm{E}\left[\!\left[u^n[k]\right]\!\right] - (n-1)!!\,\sigma^n \tag{3.27}$$

for even n, where $u^n[k]$ is a zero-mean arbitrary sequence. Referring back to the binary case, $\delta_4 = 1 - (4-1)!! = 1 - 3 = -2$, one again obtains (3.25).

Simulation experiment

A simulation experiment was performed using the MATLAB software using the structure in Fig. 3.1, with $H(z)$ set to unity resulting in a Wiener system configuration. The nonlinearity is purely cubic with unity gain, i.e. $\mathfrak{f} \mapsto x^3$, preceded by a two-tap

(first order) FIR dynamic linearity with $G(z)$ given by:

$$G(z) = 1 + 0.6\, z^{-1}\,. \tag{3.28}$$

It is imperative to note that the choice of the type, order or coefficients of the linearity filter here is arbitrary and unimportant. The theory has been confirmed to work with a wide range of linearity blocks, from FIR to IIR filters (see also p. 34). In this case, the number of FIR filter taps was chosen to be small to emphasise the discrepancy between the Gaussian and non-Gaussian BLAs—the concept of discrepancy and its relation to system memory will be introduced in Section 3.5 and Section 3.5.2 respectively.

The system was subjected firstly to discrete-interval periodic Gaussian white noise sequences of length $N = 4095$ and the corresponding BLA was obtained using (3.7), with the expectation operators implemented by the arithmetic *mean* function on auto- and cross-power spectrum estimates over 10 000 realisations. The system was secondly subjected to MLBS's also with period $N = 2^{12} - 1 = 4095$ and the BLA was obtained similarly; but instead of averaging over 10 000 realisations, 144 realisations were averaged with each having a unique MLBS as input. The reason for the reduction in averaging is that there are *only* 144 unique MLBS's available (including their mirrored counterparts) with this particular period of N (see Section 2.3.4 and the table in Appendix A.1). Since the MLBS's are deterministic unlike the Gaussian noise sequences (see Section 4.3), and there were no disturbance noises added to the system, there was no need for more averaging (using repeated realisations) to be performed.

Non-parametric estimates of the BLAs for both input types were obtained at steady-state. The dc gains of the BLAs were normalised by the same amount, so the Gaussian BLA matched that of the true linearity FRF. The (magnitude and phase) frequency responses for both input sequences are shown in Fig. 3.3, together with the FRF of the linearity, i.e. $G(z)$. Finally, the theoretical BLA for binary sequences predicted by (3.24), converted to the frequency domain, is also plotted on the same graph.

Referring to Fig. 3.3, it can be seen that the best linear frequency response estimate obtained from MLBS agrees very well with the theoretical binary BLA prediction, in terms of both the magnitude and phase frequency spectra. As mentioned in the beginning of this chapter, the Gaussian BLA is proportional to the true underlying linear dynamics $G(z)$. Hence with dc gain normalised, the results obtained from the Gaussian sequences agrees with the heavy dash-dot line indicating the true linearity $G(z)$. This simulation experiment verifies the theory predicted in this

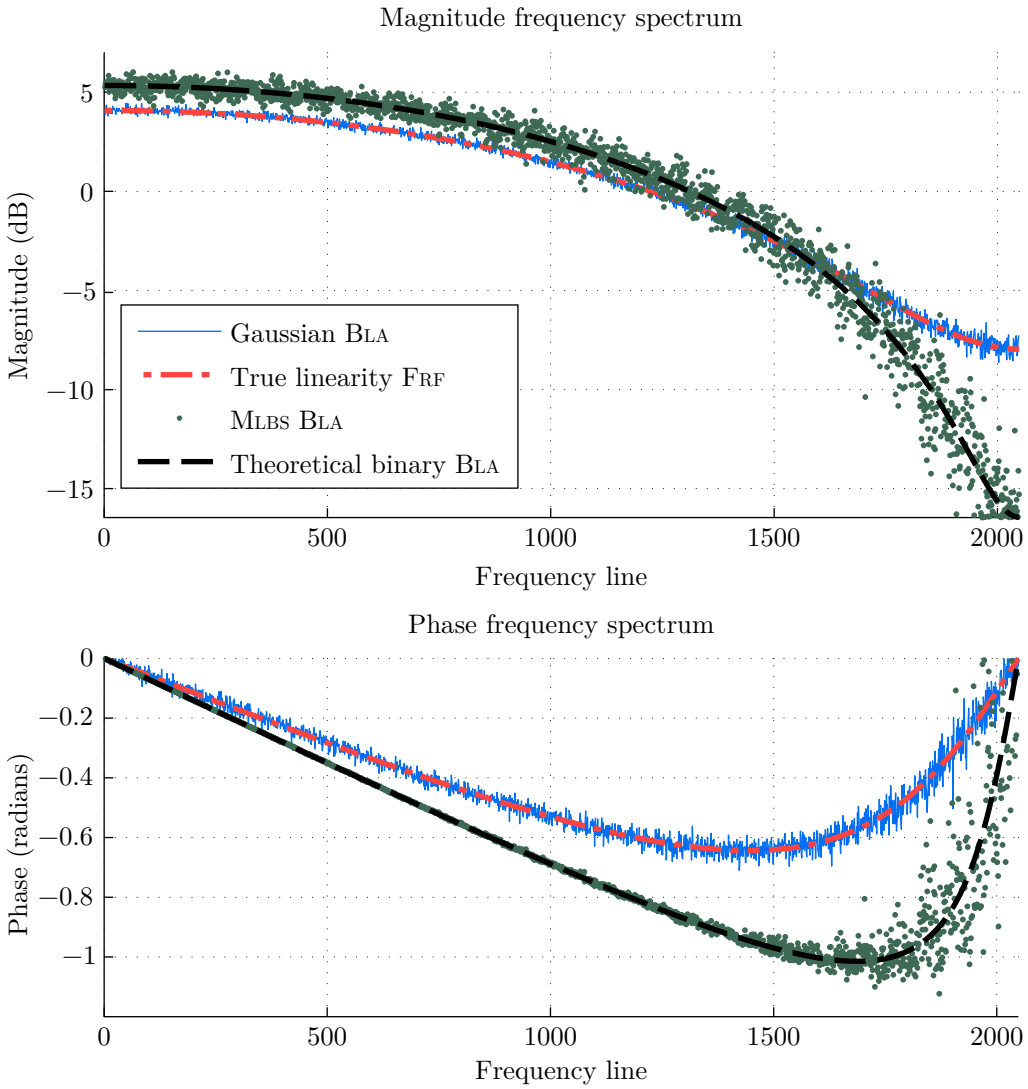


Figure 3.3: Comparison of simulation against theory: BLAs (with dc gain normalised) obtained from Gaussian and binary inputs plotted with their theoretical counterparts for a Wiener system with pure *cubic* nonlinearity

section.

3.3.2 Pure quintic nonlinearity

For both zero mean Gaussian sequences with variance σ^2 and zero mean binary sequences with levels $\pm V$, if power is normalised to unity then $\sigma^2 = V^2 = 1$. Using the results from Appendix A.3, for a Wiener system with pure quintic nonlinearity $\mathfrak{f} \mapsto x^5$, the BLAs are given by (A.3.2) for the Gaussian input case, where:

$$g_{\mathcal{N}\,\mathrm{BLA}}[r] = 15\alpha_2^2 g[r] \tag{3.29}$$

and (A.3.8) for the binary input case, where:

$$g_{\mathcal{U}_2\,\mathrm{BLA}}[r] = g_{\mathcal{N}\,\mathrm{BLA}}[r] + 16g^5[r] - 20\alpha_2 g^3[r] - 10\alpha_4 g[r]. \tag{3.30}$$

Here, for compactness, the norm terms α_n are defined as:

$$\alpha_n \triangleq \sum_{k=0}^{\infty} g^n[k] \tag{3.31}$$

where $g[k]$ is the IRF of the linearity.

Further, it is possible to generalise (3.30) for a sequence with an arbitrary input amplitude distribution. From (A.3.11) of Appendix A.3 this is given by:

$$g_{\mathcal{N}\,\mathrm{BLA}}[r] = 15\alpha_2^2 g[r] \tag{3.32}$$

for Gaussian inputs and

$$g_{\mathcal{X}\,\mathrm{BLA}}[r] = g_{\mathcal{N}\,\mathrm{BLA}}[r] + (\delta_6 - 15\delta_4)g^5[r] + 10\delta_4\alpha_2 g^3[r] + 5\delta_4\alpha_4 g[r] \tag{3.33}$$

for arbitrary inputs. In the above equation, the non-Gaussian moment correction terms δ_n are defined as before in (3.27), i.e. $\delta_n \triangleq \mathrm{E}\left[\!\left[u^n[k]\right]\!\right] - (n-1)!!\,\sigma^n$ for even n and $u[k]$ is the arbitrary input.

As one can see, comparing to the pure cubic nonlinearity case in the previous section where (3.25) can be rewritten as:

$$g^{(3)}_{\mathcal{U}_2\,\mathrm{BLA}}[r] = g^{(3)}_{\mathcal{N}\,\mathrm{BLA}}[r] - \delta_2 g^3[r], \tag{3.34}$$

the quintic BLA of (3.30) is much more complicated. If the cubic term is also present in the nonlinearity, the BLA is a superposition of (3.34) and (3.30).

Note that given a measurement of $g_{\mathcal{X}\,\mathrm{BLA}}$, it is not feasible to eliminate the discrepancy by solving for $g_{\mathcal{N}\,\mathrm{BLA}}$ in (3.30) or (3.34) by varying the input amplitude for example, as doing so requires the explicit knowledge of the linearity and the nonlinearity (which may contain a mixture of degrees). Having the knowledge of

the linearity defeats the purpose as $g_{\mathcal{N}\,\mathrm{BLA}}[r]$ is known to be proportional to $g[r]$ (see (3.12)).

Simulation experiment

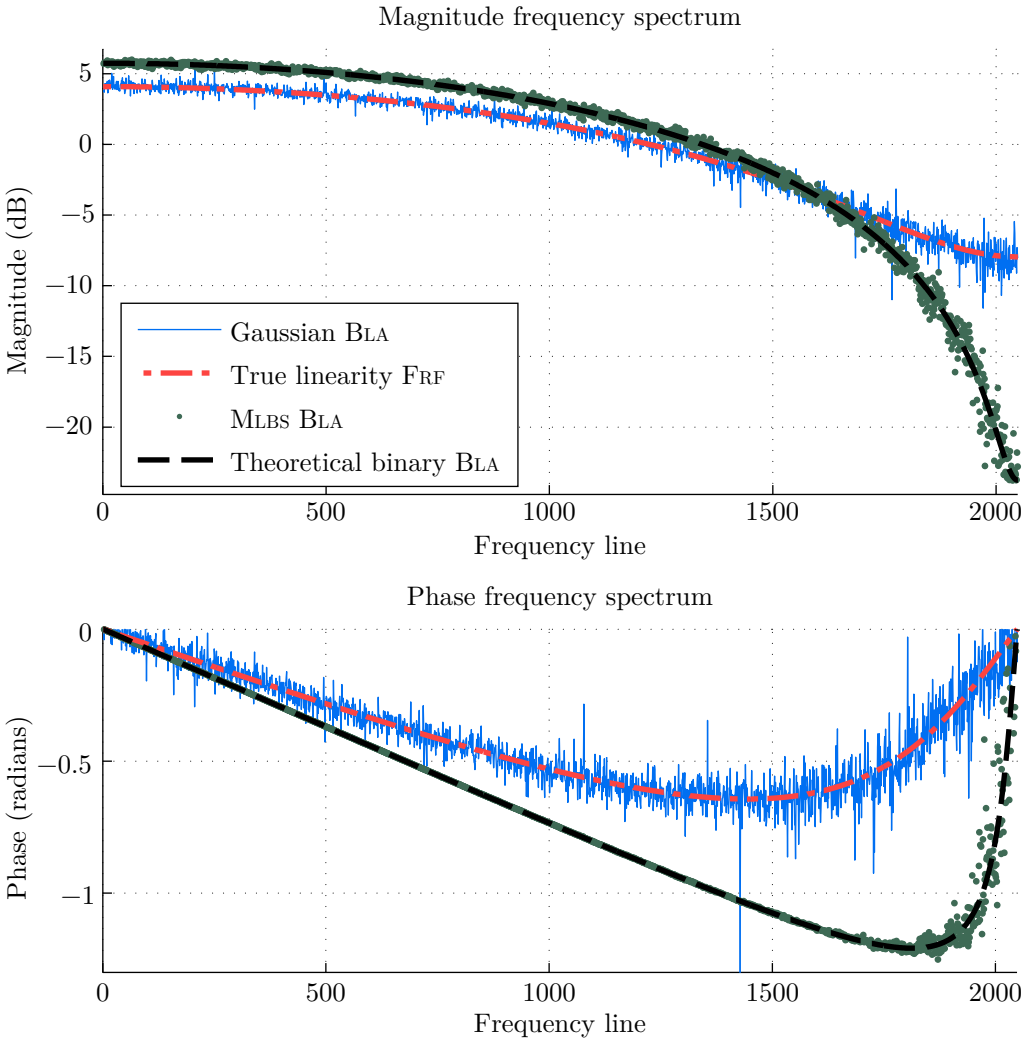


Figure 3.4: As Fig. 3.3, with pure quintic instead of cubic nonlinearity

A simulation experiment with the same setup as that outlined near the end of Section 3.3.1, but with a pure quintic nonlinearity replacing the pure cubic, was performed to verify the theory with results plotted in Fig. 3.4. Similar to the cubic case, there is an excellent agreement between the theoretical and the estimated BLAs.

As verification of the theory, physical experiments performed on an electronic discrete-time Wiener system with a pure cubic nonlinearity will be discussed in Chapter 6.

3.4 Discrete-time Volterra systems

So far the theory has been developed for a WH structure. The theory can be generalised to discrete-time Volterra systems that encompass a much wider range of time-invariant nonlinear systems (Volterra, 1930/2005). If a nonlinear system contains feedback, the theory in this section should provide an adequate approximation if the input-output relationship of the said system can be represented sufficiently well by a Volterra series of a finite degree.

Following on from the Stone-Weierstrass theorem (Stone, 1948), any time-invariant, finite memory, causal, discrete-time nonlinear system that is a continuous functional[1] of the input can (over a uniformly bounded set of input signals) be uniformly approximated to an arbitrary accuracy by a discrete-time, finite-memory Volterra series of sufficient order (Korenberg, 1991). Similar statements can also be made for continuous-time systems through the pioneering work of Fréchet (1910). Many time-invariant nonlinear dynamical systems including many block structured systems (for example, WH of Section 3.3 and parallel Wiener) can be modelled using Volterra series. The converse is also true—a parallel Wiener structure, for example, may represent any Volterra systems by incorporating more and more branches within any arbitrary level of accuracy (Westwick & Kearney, 2003). Hence, BLA theory applicable to Wiener systems can be, by principle of superposition, applied to a Volterra System.

Here, the theoretical BLA is derived for white zero-mean Gaussian and binary sequences for a generic discrete-time Volterra system (see Appendix A.2 for the assumptions made). The zero-mean requirement is not a big deal in practice, as a dc term simply changes the operating point at which the system is linearised. Discrepancies between the Gaussian and binary BLA however, depend on the power spectra and the higher order moments of both types of inputs. Hence, both the Gaussian and binary sequences are assumed to be white and have power normalised to unity.

A SISO Volterra system is represented by Fig. 3.5. Its output is given by a Volterra series, which contains a sum of multi-dimensional convolution integrals of

[1]The term 'functional' in mathematics refers to a function which maps a vector space to a scalar field; in this case, from the vector of the system input in different past time points to the output at a given time.

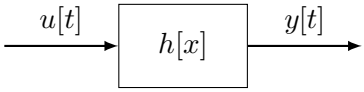


Figure 3.5: Generic Volterra system structure

the system input with the Volterra kernels, from degree of one to infinity. A Volterra kernel may be considered a multi-dimensional analogue of an IRF. The zeroth kernel contribution $y^{(0)}$ is a constant independent of system input. The first degree of the Volterra kernel $h^{(1)}[k]$ with a dimension of one (containing a single argument) is known as the linear kernel and this is equivalent to the IRF $g[k]$ for a linear system. A purely linear system has the linear kernel equal to its IRF and all higher order kernels are exactly zero; as such, a linear dynamical system is a degenerate case of a Volterra system. For nonlinear systems, one or more higher order kernels are non-trivial, i.e. having non-zero output contributions. These multi-dimensional kernels capture the nonlinear effects. Given a time-invariant nonlinear system depicted in Fig. 3.5, the output y is an aggregate sum of contributions from an infinite number of Volterra kernels such that $y = \sum_{q=0}^{\infty} y^{(q)}$, where $y^{(q)}$ is the output contribution from a single kernel of order q. $y^{(q)}$ for $q \geq 1$ is given by:

$$y^{(q)}[t] = \sum_{k_1=0}^{\infty} \cdots \sum_{k_q=0}^{\infty} h^{(q)}\left[k_1, \ldots, k_q\right] \prod_{\alpha=1}^{q} u[t-k_\alpha] \tag{3.35}$$

where $h^{(q)}$ is the q^{th} Volterra kernel. Without loss of generality, normalising the power of the input to unity and using (3.9), the BLA for the q^{th} kernel is given by:

$$\begin{aligned} g_{\text{BLA}}^{(q)}(r) &= \mathrm{E}\left[\!\left[y^{(q)}[t] u[t-r] \right]\!\right] \\ &= \sum_{k_1=0}^{\infty} \cdots \sum_{k_q=0}^{\infty} h^{(q)}\left[k_1, \ldots, k_q\right] \mathrm{E}\left[\!\left[u[t-r] \prod_{\alpha=1}^{q} u[t-k_\alpha] \right]\!\right]. \end{aligned} \tag{3.36}$$

The law of superposition applies since if $y = \sum_{q=1}^{\infty} y^{(q)}$, $g_{\text{BLA}} = \sum_{q=1}^{\infty} g_{\text{BLA}}^{(q)}$.

The mathematical work for deriving the theoretical contributions from an arbitrary Volterra kernel degree to the BLA is shown in Appendix A.2. Based upon this, the theoretical contributions from the 3^{rd} and 5^{th} degree Volterra kernels are given in the next two sections.

3.4.1 Third degree Volterra contributions

The following results apply to time-invariant discrete-time Volterra systems excited by zero-mean inputs with normalised power of unity.

Gaussian input case

Appendix A.2 has shown that for zero-mean white Gaussian inputs, (3.36) may be simplified to (A.2.10), reproduced as follows:

$$g^{(q)}_{\text{BLA}}[r] = q!! \sum_{k_1}^{\infty} \cdots \sum_{k_p}^{\infty} h^{(q)}\left[k_1, k_1, k_2, k_2, \ldots, k_p, k_p, r\right] \tag{3.37}$$

where q is odd and $p = (q-1)/2$. From this equation, the BLA of a nonlinear system described by a 3rd degree Volterra kernel, excited by zero-mean Gaussian inputs with normalised power, is given by:

$$\begin{aligned} g^{(3)}_{\mathcal{N}\,\text{BLA}}[r] &= 3!! \sum_{k}^{\infty} h^{(q)}\left[k, k, r\right] \\ &= 3 \sum_{k}^{\infty} h^{(q)}\left[k, k, r\right]. \end{aligned} \tag{3.38}$$

Binary input case

With respect to zero-mean binary inputs, using (A.2.14), one can deduce that a 3rd degree Volterra series contains the following forms of non-trivial (non-zero) contributions:

$$\begin{aligned} &\begin{cases} \binom{3}{3}\, h^{(3)}\left[r, r, r\right] & \text{for } k = r \\ \binom{3}{2,1}\, h^{(3)}\left[k, k, r\right] & \text{for } k \neq r \end{cases} \\ &= \begin{cases} h^{(3)}\left[r, r, r\right] & \text{for } k = r \\ 3h^{(3)}\left[k, k, r\right] & \text{for } k \neq r. \end{cases} \end{aligned} \tag{3.39}$$

The binary BLA is then given by:

$$g^{(3)}_{\mathcal{U}_2\,\text{BLA}}[r] = h^{(3)}\left[r, r, r\right] + 3 \sum_{\substack{k=0 \\ k \neq r}}^{\infty} h^{(3)}\left[k, k, r\right].$$

Incorporating the first term into the second term to remove the restriction of $k \neq r$ in the summation yields:

$$\begin{aligned} g^{(3)}_{\mathcal{U}_2\,\text{BLA}}[r] &= -2h^{(3)}\left[r, r, r\right] + 3 \sum_{k=0}^{\infty} h^{(3)}\left[k, k, r\right] \\ &= g^{(3)}_{\mathcal{N}\,\text{BLA}}[r] - 2h^{(3)}\left[r, r, r\right]. \end{aligned} \tag{3.40}$$

Note that if the nonlinear system is a Wiener system constructed from a linearity with IRF of $g[k]$ followed by a pure cubic nonlinearity, its Volterra kernel

$h[a,b,c]$ collapses into the product $g[a]g[b]g[c]$. In such a case, (3.40) is simplified to the previous result of (3.25), as required.

Arbitrary input case

Similar to the steps performed in deriving (3.39), for any zero-mean arbitrary inputs, there are non-trivial contributions of the Volterra series in the form of:

$$\begin{cases} \binom{3}{3}\, h^{(3)}\,[r,r,r]\,\mathfrak{M}_4 & \text{for } k=r \\ \binom{3}{2,1}\, h^{(3)}\,[k,k,r]\,(\mathfrak{M}_2)^2 & \text{for } k\neq r \end{cases} \tag{3.41}$$

where $\mathfrak{M}_n \triangleq \mathrm{E}[\![u^n]\!]$ is the moment notation defined in Section 3.2.

Recall from (3.27) that the definition of the non-Gaussian moment correction terms is $\delta_n \triangleq \mathrm{E}\left[\!\left[u^n[k]\right]\!\right] - (n-1)!!\,\sigma^n$ for even n, where $u^n[k]$ is some zero-mean arbitrary sequence. Normalising the power of the sequences such that $\sigma^2 = \mathfrak{M}_2 = \mathrm{E}\left[\!\left[u^2[k]\right]\!\right] = 1$, one can write the contributions to BLA of this arbitrary sequence in terms of Gaussian moment by applying (A.3.9) for both Gaussian and arbitrary signals as:

$$\begin{aligned} g^{(3)}_{\mathcal{X}\,\mathrm{BLA}} &= g^{(3)}_{\mathcal{N}\,\mathrm{BLA}} + g^{(3)}_{\mathcal{X}\,\mathrm{BLA}} - g^{(3)}_{\mathcal{N}\,\mathrm{BLA}} \\ &= g^{(3)}_{\mathcal{N}\,\mathrm{BLA}} + \delta_4 h^{(3)}\,[r,r,r] + (3-3)\cancel{\sum_{k=0}^{\infty} h^{(3)}\,[k,k,r]} \\ &= g^{(3)}_{\mathcal{N}\,\mathrm{BLA}} + \delta_4 h^{(3)}\,[r,r,r]. \end{aligned} \tag{3.42}$$

If the nonlinear system is a Wiener system with IRF of $g_{\mathrm{L}}[k]$ followed by a pure cubic nonlinearity, (3.42) becomes:

$$g_{\mathcal{X}\,\mathrm{BLA}}[r] = g^{(3)}_{\mathcal{N}\,\mathrm{BLA}}[r] + \delta_4 g^3_{\mathrm{L}}[k]. \tag{3.43}$$

If the input is binary, using the definition of δ_n, $\delta_4 = 1-(4-1)!! = 1-3 = -2$. Substituting these into (3.43) one obtains again (3.25) as expected.

3.4.2 Fifth degree Volterra contributions

The results for the 5^{th} Volterra kernel are reproduced from similar workings shown in Appendix A.3, up until where the expressions are adapted for Wiener systems. These results apply to time-invariant discrete-time Volterra systems excited by zero-mean white inputs with normalised power of unity.

Gaussian input case

From (A.3.1):

$$g^{(5)}_{\mathcal{N}\,\mathrm{BLA}}[r] = 15\sum_{k_1}^{\infty}\sum_{k_2}^{\infty} h^{(q)}\left[k_1, k_1, k_2, k_2, r\right]. \tag{3.44}$$

Binary input case

From (A.3.6):

$$\begin{aligned} g^{(5)}_{\mathcal{U}_2\,\mathrm{BLA}}[r] = g^{(5)}_{\mathcal{N}\,\mathrm{BLA}} &- 20\sum_{k=0}^{\infty} h^{(5)}\left[k, k, r, r, r\right] \\ &- 10\sum_{k=0}^{\infty} h^{(5)}\left[k, k, k, k, r\right] + 16h^{(5)}\left[r, r, r, r, r\right]. \end{aligned} \tag{3.45}$$

Arbitrary input case

From (A.3.10):

$$\begin{aligned} g^{(5)}_{\mathcal{X}\,\mathrm{BLA}} = g^{(5)}_{\mathcal{N}\,\mathrm{BLA}} &+ 10\delta_4\sum_{k=0}^{\infty} h^{(5)}\left[k, k, r, r, r\right] \\ &+ 5\delta_4\sum_{k=0}^{\infty} h^{(5)}\left[k, k, k, k, r\right] \\ &+ (\delta_6 - 15\delta_4)h^{(5)}\left[r, r, r, r, r\right]. \end{aligned} \tag{3.46}$$

3.5 The Discrepancy Factor

Definition Differences arise between the BLA estimated using sequences with Gaussian amplitude distribution and that obtained using sequences with other amplitude distributions. The amplitude distribution of an input is defined by the probability density function (p.d.f.) for a continuous-level signal or the probability mass function (p.m.f.) for a discrete-level sequence. If the Gaussian BLA is used as a reference, the discrepancy between the two can be quantified as the sum squared errors, normalised by the total energy of the Gaussian BLA. This measure shall be called the Discrepancy Factor (DF).

In the frequency domain, for the Gaussian BLA estimate $\hat{G}_{\text{Gaussian BLA}}$ and some other BLA estimated with an input sequence of some other distribution $\hat{G}_{\text{Arbitrary BLA}}$, the *empirical* Discrepancy Factor (DF) denoted by Fraktur script

$\mathfrak{D}$ is defined as follows:

$$\mathfrak{D} \triangleq \frac{\sum_{k=0}^{N-1}\left|\hat{G}_{\text{Arbitrary BLA}}[k] - \hat{G}_{\text{Gaussian BLA}}[k]\right|^2}{\sum_{k=0}^{N-1}\left|\hat{G}_{\text{Gaussian BLA}}[k]\right|^2}. \tag{3.47a}$$

Using Parseval's theorem, the time domain equivalent can be similarly defined as:

$$\mathfrak{D} \triangleq \frac{\sum_{k=0}^{N-1}\left|\hat{g}_{\text{Arbitrary BLA}}[k] - \hat{g}_{\text{Gaussian BLA}}[k]\right|^2}{\sum_{k=0}^{N-1}\left|\hat{g}_{\text{Gaussian BLA}}[k]\right|^2}. \tag{3.47b}$$

One can deduce the *theoretical* DF by observing that its definition as a ratio of power between the contributions, from the discrepancy terms to the Gaussian terms of the BLA. For a Wiener system with a pure *cubic* nonlinearity and an arbitrary input, the theoretical DF is the ratio between the sums of the squares of the first and second terms on the right hand side of (3.26), so that:

$$\mathfrak{D}^{(3)}_{\text{theory, }\mathcal{X}} = \frac{\cancel{p_3^2}\delta_4^2 \sum_{r=0}^{\infty}\left(g_r^3\right)^2}{9\cancel{p_3^2}\sigma^8 \sum_{r=0}^{\infty} g_r^2 \left(\sum_{k=0}^{\infty} g_k^2\right)^2},$$

normalising the Gaussian input power to unity such that σ^8=1, one obtains:

$$= \frac{\delta_4^2 \sum_{k=0}^{\infty} g_k^6}{9\left(\sum_{k=0}^{\infty} g_k^2\right)^3} = \frac{\delta_4^2 \alpha_6}{9\alpha_2^3} \tag{3.48}$$

where δ_n and α_n, respectively, are as defined before in (3.27) and (3.31). If the arbitrary input is *binary*, using the previous result for the binary case, $\delta_4 = -2 \Rightarrow \delta_4^2 = 4$, (3.48) then becomes:

$$\mathfrak{D}^{(3)}_{\text{theory, }\mathcal{U}_2} = \frac{4\alpha_6}{9\alpha_2^3}. \tag{3.49}$$

For a pure *quintic* nonlinearity, the theoretical DF can be shown (Wong et al., 2012a) to be:

$$\mathfrak{D}^{(5)}_{\text{theory, }\mathcal{U}_2} = \frac{\left(\epsilon_5\alpha_5 + \epsilon_3\alpha_3\alpha_2 + \epsilon_1\alpha_1\alpha_4\right)^2}{225\alpha_2^5} \tag{3.50}$$

where ϵ_5, ϵ_3 and ϵ_1 are factors of additive bias, which arise from non-Gaussian moment deviations corresponding the coefficients of the fifth, third and first powers of the filter coefficients along r. For a quintic nonlinearity and zero mean binary inputs, specifically $\epsilon_5 = 16$, $\epsilon_3 = -20$ and $\epsilon_1 = -10$ (see Appendix A.3).

3.5.1 Simulation experiment

To verify that the theoretical expression given by (3.49) agrees with the empirical DF calculated from (3.47), two different sets of Wiener systems were considered, one with the linearity having first-order dynamics with transfer function:

$$G(z) = \frac{1}{z - e^{-\gamma}} \tag{3.51}$$

and the other having second order dynamics with transfer function:

$$\begin{cases} G(z) = \frac{1}{(z-p)(z-\bar{p})} \\ p = e^{-\gamma}(\cos\frac{\pi}{12} \pm \mathrm{j}\sin\frac{\pi}{12}) \end{cases} \tag{3.52}$$

where $\bar{p}$ is the complex conjugate of p. In both cases, the nonlinearity was a pure cubic function $\mathfrak{f}(x) \mapsto x^3$. A range of values of time constants $1/\gamma$ between 0.1 and 50 was used to vary the length of the linearity IRF $g[k]$ and this in turn affects the DFS (the effect is discussed later in Section 3.5.2). Periodic Gaussian noise and DIRBS's, both with length of 1024 and with levels ± 1 (such that in both case the input power equals unity), were used as inputs. The sampling time was chosen to be 1 second for simplicity. The results were obtained once the system had reached steady-state. Non-parametric estimates of the BLAS were first obtained from direct division of input-output cross-spectrum by input power spectrum as in (3.3), averaged over 10 000 realisations (prior to the division).

To reduce the variances of the non-parametric estimates, parametric models of the BLA were then obtained by feeding the non-parametric frequency response estimates into an iterated nonlinear weighted least squares procedure provided by ELiS (see glossary entry). The variances of the complex frequency response estimates were also supplied to ELiS for weighting purposes. The cost function used internally was $C = \sum_{k=0}^{N-1} W_k \left|e_k\right|^2$ where $\left|e_k\right|^2$ are the squared errors between modelled output and actual output in terms of frequency line number k, and W_k are the weighting factors proportional to the reciprocal of the variance at each k. The DFS for the estimated parametric models were calculated using (3.47). These values are plotted in Fig. 3.6 against their theoretical counterparts found from the true linearity coefficients using (3.49). It can be seen that there is a well matched one-to-one relationship between theory and simulation.

3.5.2 Effect of system memory

Generally speaking, estimating the BLA of a system with a short memory, for example, expressed by an IRF having a length in an order of tens of samples, results in

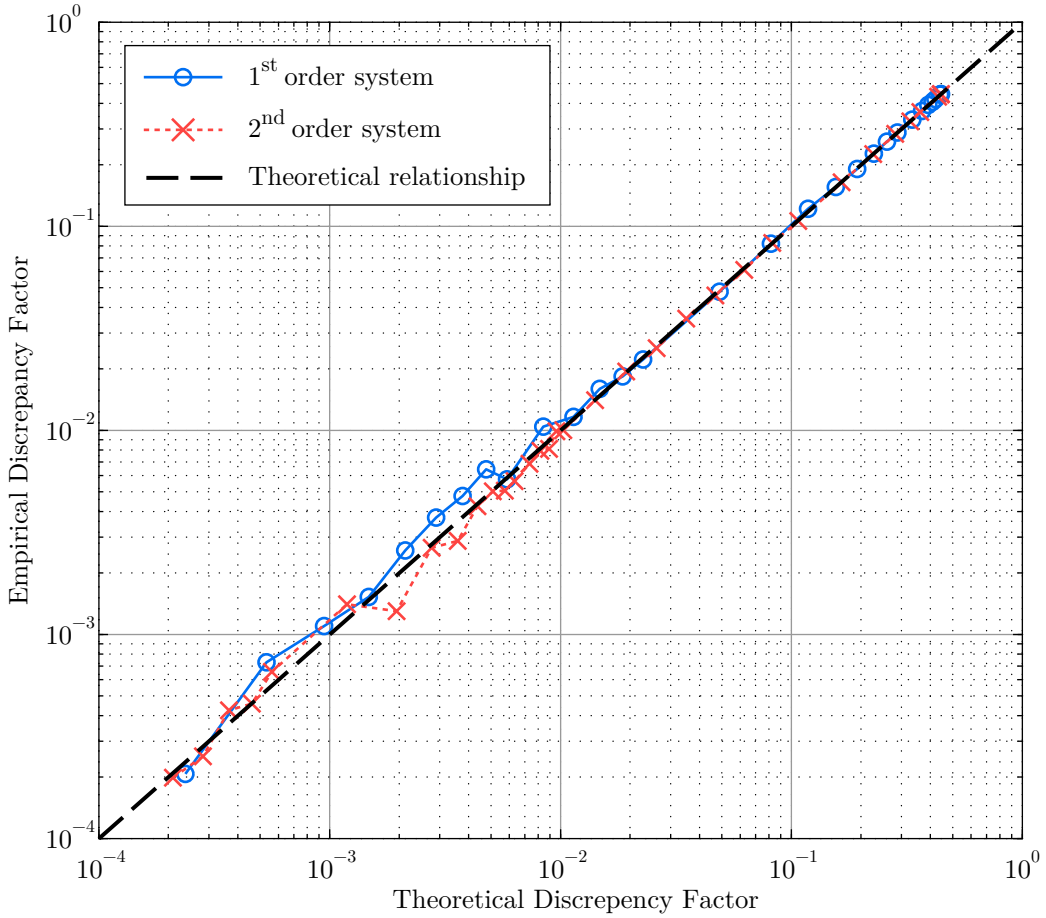


Figure 3.6: *Empirical* (measured) Discrepancy Factors obtained from random binary sequences and Gaussian noise against *theoretical* Discrepancy Factors for various Wiener systems with pure cubic nonlinearity (simulation)

a high DF. Having verified the equivalence of the empirical DFs with the theoretical counterparts in the previous section, the effect of the theoretical DF in terms of length of the IRF is investigated through the experiment setup of both the first and second order Wiener systems given in the previous section.

For the first order Wiener system with a linearity given by (3.51) preceding a pure cubic nonlinearity, the time constant is varied from 0.1 second to 500 seconds with 50 equidistant points in logarithmic space. This results in a *1% settling time*, taken as time taken to reach 1% of the initial value found using (A.4.2), between $-0.1 \ln 0.01 = 0.461\,\text{s}$ and $-500 \ln 0.01 = 2.30 \times 10^{3}\,\text{s}$. The theoretical result is shown in Fig. 3.7. The 1% settling time used in the x-axes of the subsequent graphs are obtained for the first and second order systems using procedures detailed in Appendix A.4. It can be seen from Fig. 3.7 that a shorter impulse response results

in a larger DF. The maximum limit of the DF for a pure cubic nonlinearity is eactly $4/9 = 44.4\%$ to 3 significant figures. This is certainly not negligible and calls for a method to minimise the DF—one such method will be discussed in Chapter 5.

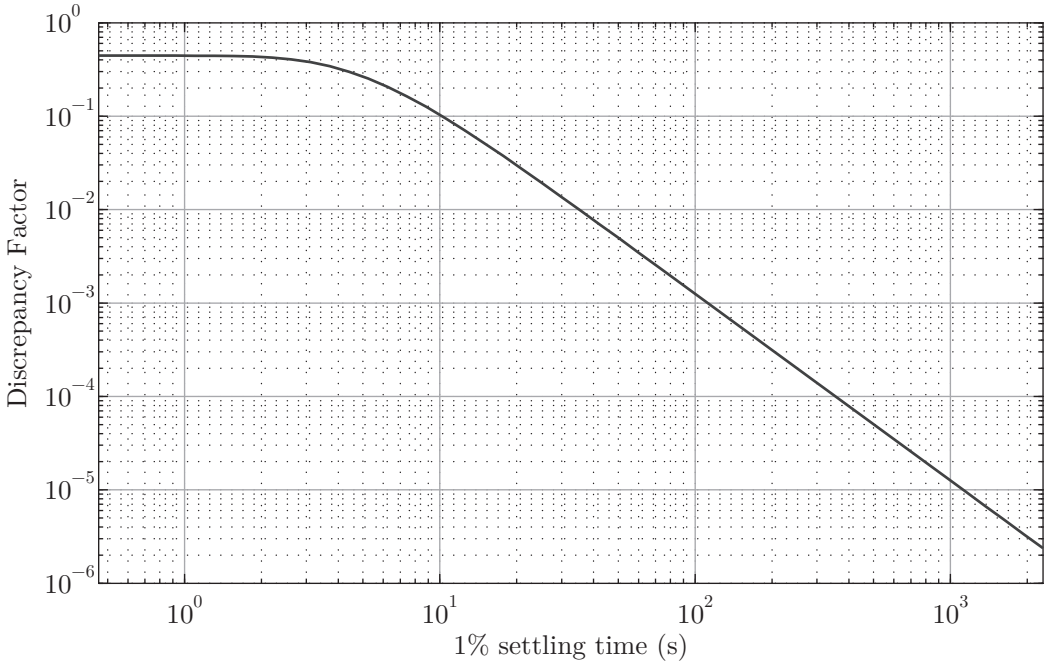


Figure 3.7: Discrepancy factor as a function of settling time of a first order linearity in a Wiener system

To illustrate that this effect is not limited to a first order linearity, a second order Wiener system is considered with its linearity given by (3.52) with a pure cubic nonlinearity. 50 conjugate pairs of poles were tried, with γ being 50 equidistant points in logarithmic space from 2×10^{-5} to 1. The independent variable γ affects the settling time directly (see Appendices A.4 and A.5), resulting in a 1% settling time found by (A.4.4) between $-\ln 0.01 = 4.61\,\text{s}$ and $-(\ln 0.01)/(2 \times 10^{-5}) = 2.30 \times 10^{5}\,\text{s}$. The range of γ tested results in the poles configurations in the z-domain as illustrated by the z-plane pole-zero plot of Fig. 3.8. In the figure, the contours marked by decimal numbers denote regions of constant damping factor ζ, while the contours containing π denote regions of constant undamped natural frequency ω_{n}. The DFs as a function of different pole configurations of this set of 2^{nd} order systems are shown in Fig. 3.9.

One can see the general trend that the shorter the memory of the linearity, the higher the DF is between the binary and Gaussian BLAs. To illustrate that this applies to a more complex system, a simulation experiment was performed for a 3-branch additive parallel Wiener structure, in which the three linearities were all

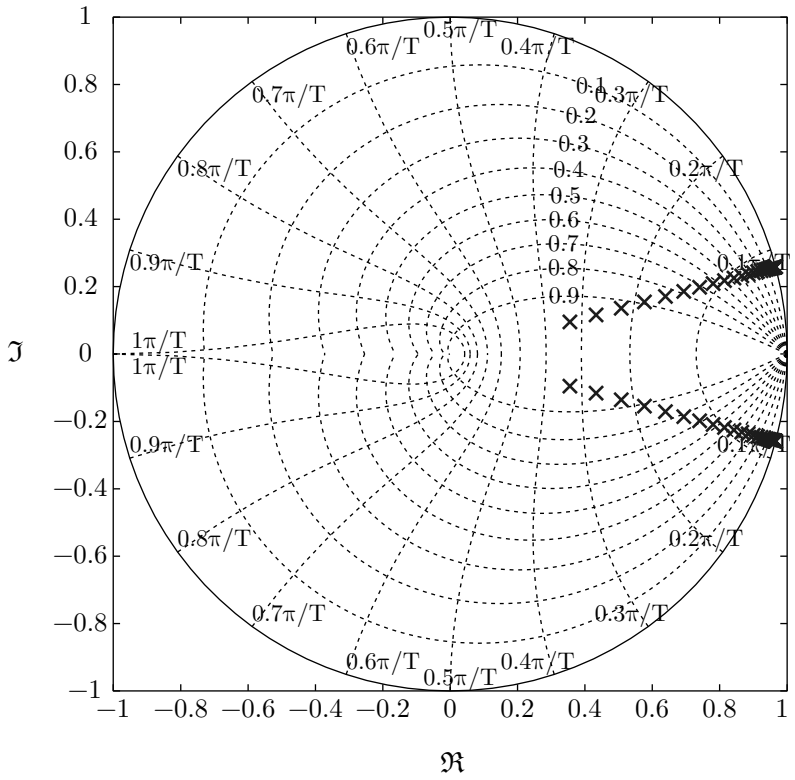


Figure 3.8: z-plane plot of the various poles configurations of the second order linearity used in Section 3.5.2

first order filters with IRF g_1, g_2 and g_3. The settling times of g_1 and g_2 were varied by changing the time constants of the linearities in two branches. The values used were the same as those used in Fig. 3.7. The settling time of g_3 was fixed to either of the extremes of 2.30×10^4 s or 0.461 s. Results of this setup are shown in Fig. 3.10

The minimum DF is obtained when all g_1, g_2 and g_3 have long memory, as depicted in Fig. 3.10a. The opposite is also true, when all filters have short memory, a maximum DF is reached as shown in Fig. 3.10b, wherein a miniature sub-figure on the top left hand corner shows the same graph on an equal axis scaling as that of Fig. 3.10a. It can be seen that all filters should have long memory in order to have a low DF. This tells us that the requirement resides on the overall memory of the combined linear dynamics rather than individual linearity elements. Other structures not presented here, such as multiplicative Wiener branches, were seen to be consistent with this general conclusion.

From a practical point of view, when identifying nonlinear systems with inputs restricted to binary sequences, the clock frequency should be chosen such that the time constant of the predominant dynamic should be much greater than

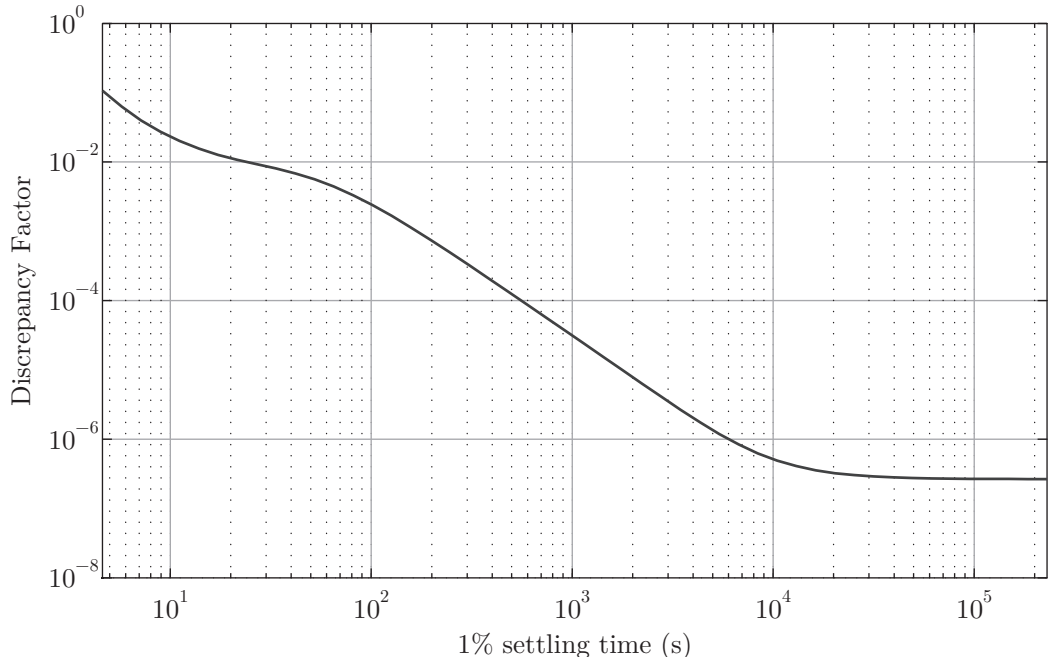
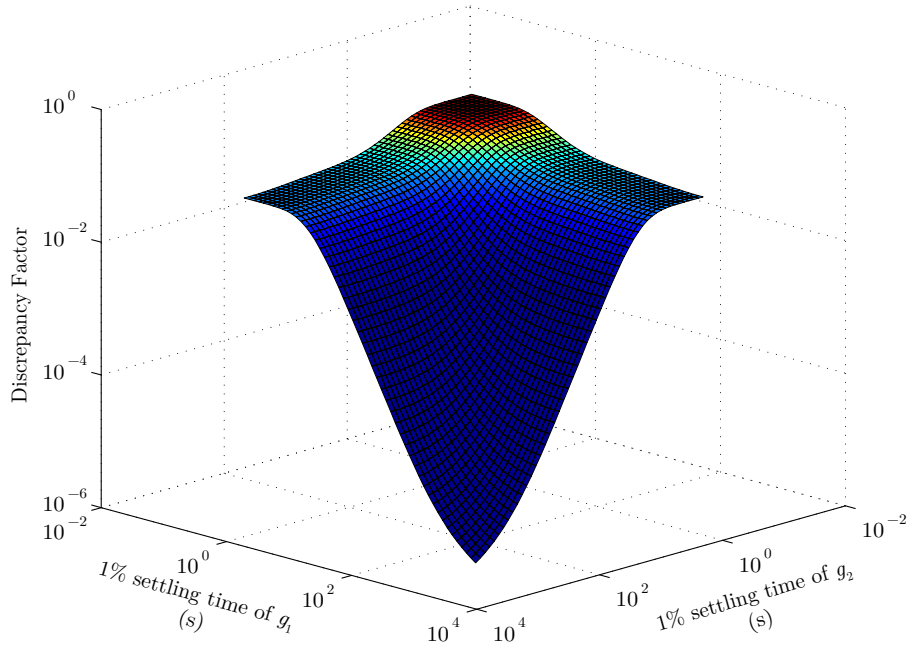


Figure 3.9: Discrepancy factor as a function of the settling time of a second order linearity in a Wiener system

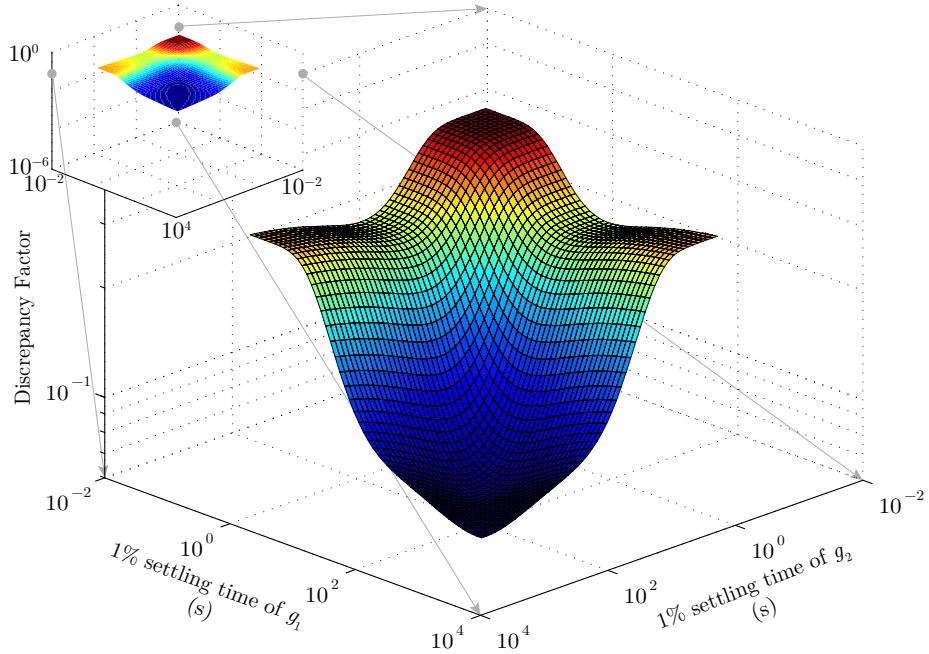
the sequence interval in order to limit the DF. Precisely how much greater depends on the system in question, which the user may have no prior knowledge. Nevertheless estimates of the DF can be made through a method described in the next section. Adjusting the clock frequency is not always possible or desirable as it is limited by the signal generator or how fast an actuator can change when subject to an abrupt zero-order hold (ZOH) input signal—in which case increasing the clock frequency may introduce extra dynamics or nonlinearities from the actuator. The dynamics may be incorporated into the model or separated using indirect system identification techniques by utilising information on both the measured input and reference input (Pintelon & Schoukens, 2012, eq. 2.64). Additional nonlinear effects however may be difficult to tackle and have to be accounted for.

3.5.3 Checking the level of bias for non-Gaussian inputs

From the definition in (3.47), the DF is a measure of the bias of BLAS obtained using non-Gaussian input signals rather than Gaussian. Direct calculation to yield the *empirical (measured)* DF requires the Gaussian BLA, while the calculation of the *theoretical* DF requires the true IRF of the underlying system. The objective is to quantify the discrepancy without invoking the use of Gaussian signals—since



(a) g_3 with a long settling time



(b) g_3 with a short settling time

Figure 3.10: Discrepancy Factor of a cubic nonlinearity 3-branch parallel Wiener system with linearities g_1 and g_2 of various settling time; and that of g_3 was fixed at either (a) 2.30×10^4 s or (b) 0.461 s. Miniature figure in (b) shows (b) with the same axes scaling as that of (a).

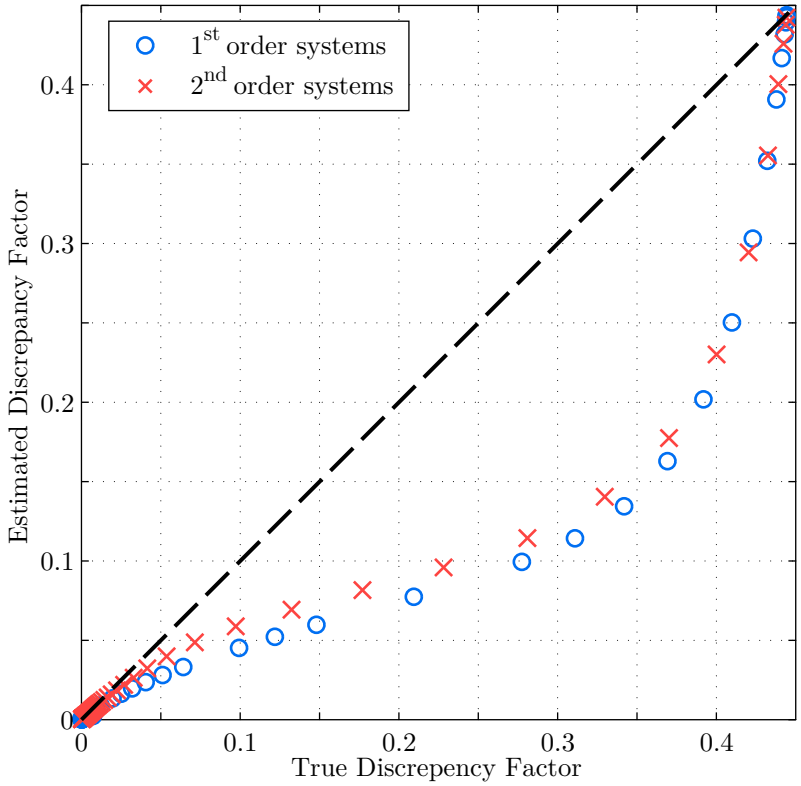


Figure 3.11: The estimated and true Discrepancy factors in various Wiener first and second order systems

the system may not be suitable for the use of Gaussian due to actuator constraints. It is possible to obtain a rough estimate of the DF by substituting the underlying linearity IRF for the non-Gaussian BLA estimate in the equations of the *theoretical* DF to obtain an estimate for the true DF. This could give an indication on the quality of the measurements in terms of the amount of bias in practice.

A theoretical (as opposed to numerical) simulation exercise was constructed to investigate the validity of such a claim. The systems considered were identical to those in Section 3.5.2, with binary and Gaussian inputs being compared. Having verified the equivalence between the empirical and theoretical DFS in Section 3.5.1, the DFS were calculated theoretically rather than empirically. From the setup, the binary BLAS and the *true* DFS were calculated for each of the first and second order systems—the binary BLAS was calculated using (3.24) with g being the IRF of the underlying linearity, and the DFS were calculated using (3.49). The *estimated* DFS were also obtained using (3.49) for each system, but instead of using the underlying linearity as g, one assumes the true linearity was unknown and the theoretical binary BLA of $g_{\mathcal{U}_2\,\mathrm{BLA}}$ was used. As such, one does not expect these *estimated* DFS to be

equal to the *true* DFs. The *true* and *estimated* DFs are compared in Fig. 3.11.

It can be seen that the estimated DF values are understating the true discrepancy between the Gaussian 'unbiased' BLA and non-Gaussian 'biased' BLA. Nevertheless, at DFs up to 0.05, the estimated DFs agree reasonably closely with the actual DFs. This is expected; when the DF is small, the IRF estimate of the binary case would be similar to one obtained from Gaussian and hence the theoretical BLA obtained using either of them would be similar to each other. For values higher than 0.05, there is significant deviation of the estimated dynamics from those of the true underlying system. This supports the previous claim that this approximate calculation may be useful as a quick check to be performed in situations where non-Gaussian inputs were used in nonlinear system identification. At the top right hand corner of Fig. 3.11, both the true and estimated DFs approaches the natural upper limit of $^4/_9$ as evident in (3.49).

3.6 Conclusions

This chapter has shown how a discrepancy arises between the Best Linear Approximations (BLAs) obtained from Gaussian sequences with other arbitrary sequences such as binary. For the first time, theoretical expressions applicable to arbitrary inputs with respect to their amplitude distribution (or the higher order moments) of the BLA were obtained for both a discrete-time Wiener-Hammerstein (WH) system (Section 3.3) and the more general Volterra system (Section 3.4). In addition, Section 3.5 proposes a way to quantify the difference by the Discrepancy Factor (DF). The equivalence between the measured DF through simulation experiments and their theoretical counterparts was verified. Lastly, Section 3.5.2 illustrates the general trend that shorter system memory results in a higher discrepancy and Section 3.5.3 has shown how one can gauge whether the bias is likely to be high. This can be used to verify if the distribution of the input will be a critical factor for the performance of the estimation.

Chapter 4

Estimation of the BLA and the Behaviour of Nonlinear Distortions

In identification of real systems, no matter how good the experiment design and the equipment are, it is not possible to totally isolate the system from the effects of the surrounding environment. The environment may either directly or indirectly influence the actuators (inputs) exciting the system, the measurement devices (outputs) as well as the plant or process itself. Random fluctuations from the environmental sources inevitably appear as noise in the output measurements. In this thesis the output-error (OE) model or framework is used to account for these effects, this is to say, the noise is modelled as a single additive unknown component at the output and the input is assumed to be known, controlled and noise-free. The main justification for the OE framework is that a model would be derived from the reference signal (noise-free) in the computer and the measured system output (noisy).

4.1 Nonlinear distortion and exogenous noise

When taking the measurement of the output of a nonlinear system, denoted by $z(t)$, the measurement inevitably includes contributions due to exogenous environment and measurement noise (hereinafter simply referred to as noise). In the OE framework, this noise term is modelled as an additive component, denoted by $n(t)$, to the noise-free output $y(t)$. From the definition of the Best Linear Approximation (BLA)

and the relation of the BLA with the noise-free output y given by (3.5), one writes:

$$z(t) = (g_{\text{BLA}} * u)(t) + \nu(t) + n(t) \tag{4.1}$$

where ν accounts for the nonlinear distortions. In the literature, the noise n is often modelled as (filtered) additive white Gaussian noise (AWGN) and being stationary and ergodic, such that despite being a stochastic process, it has well-defined properties such as a fixed variance.

The property of the nonlinear distortion term ν depends on the input u used. Its behaviour, when the input is either random-phased multisines or Gaussian noise, has been studied by Pintelon and Schoukens (2012, p. 87), who showed that it has several properties—the most important of which are listed as follows:

1. ν has zero mean and is uncorrelated but not independent of u.
2. Given the system is periodicity-invariant (see Section 2.1), ν is also periodic with the same period as u.
3. ν is asymptotically normally distributed with a circular complex normally distributed frequency spectrum.

Because environment noise tends to be aperiodic, while the nonlinear distortions are periodic, it is possible to estimate their variances independently. This has advantages, for example, in identification algorithms that rely on the knowledge of noise variances to yield better estimates (e.g. ELiS from the MATLAB *FDIDENT* Toolbox, see their respective glossaries entries), or in enabling better modelling choices by allowing the user to ascertain the levels of nonlinear distortions. A robust method proposed in Pintelon and Schoukens (2012, sec. 4.3.1) is reproduced in the next section.

4.2 Robust method for estimating the BLA

The most widely used method in non-parametric model identification of systems in the frequency domain is to use the following relationship:

$$G(\mathrm{j}\omega) = \frac{S_{\text{ZU}}(\mathrm{j}\omega)}{S_{\text{UU}}(\omega)} \tag{4.2}$$

where $G(\mathrm{j}\omega)$ is the frequency response function, $S_{\text{ZU}}(\mathrm{j}\omega)$ is the cross power spectrum between the measured output $z(t)$ and the input $u(t)$, and $S_{\text{UU}}(\omega)$ is the auto power spectrum of the input (Bendat & Piersol, 1980; Rake, 1980).

For a linear system, $G(\mathrm{j}\omega)$ is estimated through averaging the power spectra over several periods or experiments to minimise the influence of output noise; for a nonlinear system however, even in the absence of environment noise, averaging

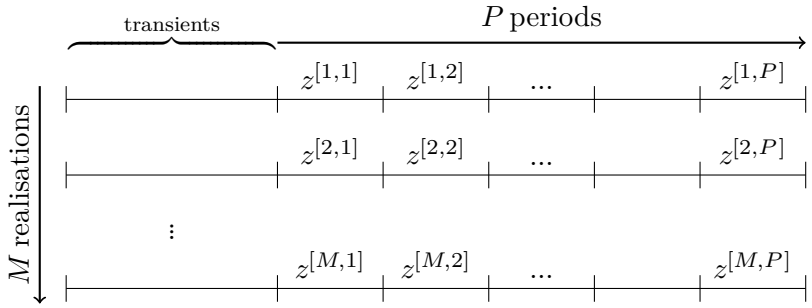


Figure 4.1: The robust procedure for estimating the BLA

must be performed to obtain $G(\mathrm{j}\omega)$ due to the dependence of ν on $\langle u \rangle$ in (4.1). To estimate $S_{\mathrm{ZU}}(\mathrm{j}\omega)$ the spectra obtained from a series of M experiments, each with a different input segment, are averaged:

$$\hat{S}_{\mathrm{ZU,M}}(\mathrm{j}\omega) = \underset{m:1\to M}{\mathrm{Avg}} \left[\!\!\left[Z^{[m]}(\mathrm{j}\omega) \cdot \overline{U}^{[m]}(\mathrm{j}\omega) \right]\!\!\right] \tag{4.3}$$

where $Y^{[m]}(\mathrm{j}\omega)$ is the discrete Fourier transform (DFT) of a period of output record of the m^{th} experiment, and $\overline{U}^{[m]}(\mathrm{j}\omega)$ is the complex conjugate of the DFT of a period of input record of the m^{th} experiment. It is known that under weak conditions (Pintelon & Schoukens, 2012, p. 50),

$$\lim_{M\to\infty} \hat{S}_{\mathrm{ZU,M}}(\mathrm{j}\omega) = S_{\mathrm{ZU}}(\mathrm{j}\omega). \tag{4.4}$$

Typically, the averaging would be performed by conducting M experiments, each using a different independent realisation of the input, over P periods of the input-output data in steady-state (i.e. after any initial transient has died away and this can be checked by looking at the correlation between inter-period data records in the output). Fig. 4.1 illustrates the averaging strategy. In the robust procedure, for each experiment m, a different periodic input realisation $u^{[m]}$ excites the system and P periods of data are collected. If the system is entirely linear and in steady-state under a periodic input, it does not matter whether one sets for example, $M = 10$ and $P = 1$, or $P = 10$ and $M = 1$. In nonlinear system identification however, the periodic nonlinear distortions can only be reduced by averaging over M realisations of different input signals, rather than over multiple periods P of the same signal.

For a nonlinear system, the global BLA estimate is given by:

$$\hat{G}_{\text{BLA}}(\mathrm{j}\omega) \triangleq \frac{\sum_{m=1}^{M}\sum_{p=1}^{P}\hat{S}_{\text{ZU}}^{[m,p]}(\mathrm{j}\omega)}{P\sum_{m=1}^{M}\hat{S}_{\text{UU}}^{[m]}(\omega)} \tag{4.5}$$

where the cross- and auto-power spectrum are $\hat{S}_{\text{ZU}}^{[m,p]} = Z^{[m,p]}\overline{U}^{[m]}$ and $\hat{S}_{\text{UU}}^{[m]} = U^{[m]}\overline{U}^{[m]}$ respectively. Note that since within each experiment the input is periodic such that $S_{\text{UU}}^{[m]}$ is not a function of the period p, there is no averaging performed across p. If the input has an invariant (fixed) auto-power spectrum (Section 4.3) across different input realisations, such as the case with, for instance, multisines with user-specified magnitude spectrum (Section 2.3.3) or pseudorandom binary sequences (PRBS's) such as Maximum Length Binary Sequences (MLBS's) (Section 2.3.4), S_{UU} is known exactly. The denominator then becomes MPS_{UU} and averaging is hence not required across m. In practice it is not advisable to assume S_{UU} would be a constant across different realisations due to nonlinear effects from signal generators or actuators. This estimator is robust against noise disturbances and is unbiased if input noise levels are small.

For known inputs, the environment and measurement noise variance σ_{n}^2 as a function of frequency ω is estimated by (Pintelon & Schoukens, 2012, eqs. 4.18 & 4.19):

$$\hat{\sigma}_{\text{n}}^2(\omega) = \frac{1}{M^2P(P-1)}\sum_{m=1}^{M}\sum_{p=1}^{P}\left|\hat{G}^{[m,p]} - \hat{G}^{[m]}\right|^2 \tag{4.6}$$

where the block estimate $\hat{G}^{[m,p]}$ and realisation estimate $\hat{G}^{[m]}$ of the BLA are defined as:

$$\hat{G}^{[m,p]} \triangleq \frac{\hat{S}_{\text{ZU}}^{[m,p]}}{\hat{S}_{\text{UU}}^{[m]}} \tag{4.7}$$

$$\hat{G}^{[m]} \triangleq \sum_{p=1}^{P}\hat{G}^{[m,p]}. \tag{4.8}$$

The total variance $\hat{\sigma}_{\text{n}+\nu}^2$, encompassing the variance introduced by nonlinear distortions ν and noise n is estimated by (Pintelon & Schoukens, 2012, eq. 4.19):

$$\hat{\sigma}_{\text{n}+\nu}^2(\omega) = \frac{1}{M(M-1)}\sum_{m=1}^{M}\left|\hat{G}^{[m]} - \hat{G}_{\text{BLA}}\right|^2 \tag{4.9}$$

It is important to note that (4.5) is not equal to the averaged (4.8) if the

input does not have invariant power spectrum across realisations. In such cases the division of the output-input cross-power spectrum (numerator) in (4.5) by the input auto-power spectrum (denominator) should be performed *after* both numerator and the denominator are averaged across m and not before. If one averages (4.8) across m this is equivalent to performing the division *before* averaging. Division is a nonlinear procedure and should be performed last, otherwise random 'dips' in the power for stochastic processes at various frequency lines can induce unacceptable variance; this is discussed in more detail in Section 4.3. Unlike stochastic sequences such as Discrete-interval Random Binary Sequence (DIRBS) (Section 2.3.2), for inputs with invariant power spectrum such as multisines and MLBS's, the denominator is simply a constant and both approaches would yield the same result. Having said that, it is still advisable to perform averaging before the division regardless of the nature of the input, due to possible nonlinear effects from generators or actuators in practice.

4.3 Stochastic and deterministic power spectra

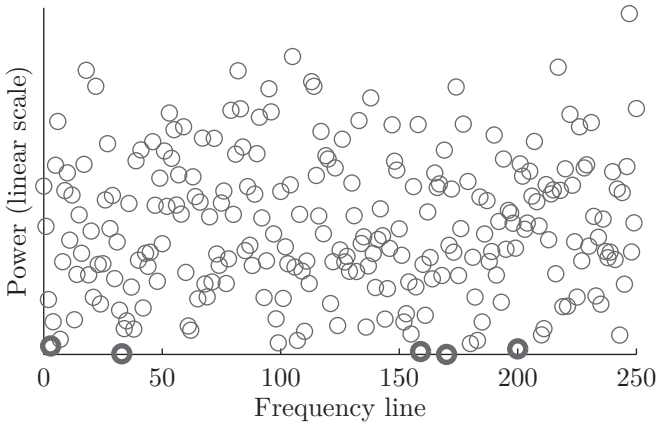


Figure 4.2: Power spectrum of a typical DIRBS of period $N = 500$ with thicker circles indicating frequencies with near zero power

Consider a family of m input signals or sequences $\{\langle u_m \rangle\}$ generated by a common statistical process $\mathcal{X}$ such that $\langle u_m \rangle \sim \mathcal{X}\ \forall m \in \{1, 2, \ldots, M\}$. If the auto-power spectra $\{S_{\mathrm{UU}}^{[m]}\}$ vary between the different realisations m, the class or family of the input is said to have *stochastic* power spectra—DIRBS and Gaussian noise sequences are examples of stochastic spectra sequences. On the other hand, if $\{S_{\mathrm{UU}}^{[m]}\}$ across m is unchanged, then the family of input is said to have a *deterministic* or an *invariant* power spectrum. For example, a family of MLBS's for a given periodicity has a deterministic power spectrum. In case of random-phased multisines, it is a

typical user choice to specify the family of multisines to have an invariant power spectrum.

In OE models with input sequences having a varying input auto-power spectra due to the stochastic nature of random signals, undesirable 'dips'—harmonics with very low power—may be present in the power spectrum. An example is given in Fig. 4.2 using a randomly generated DIRBS. Indicated by thicker circles in the figure, these dips contribute to the variance of the estimate by amplifying the output noise variance as a result of the division operation in (4.5) in the frequency lines where they occur. The dips in the averaged auto-spectra occur more frequently if smaller values of M are used.

So far, the theory and methodology mentioned in this chapter are known in the literature. It is still beneficial to verify the difference between stochastic and deterministic power spectra and to confirm that the results are similar for both continuous amplitude and discrete-level input signals; this is performed by the following set of simulation experiments:

Example A set of 500 independent simulation experiments was performed in MATLAB to showcase the identification performance difference between invariant spectrum signals and stochastic signals. Multisine (invariant spectra) and periodic Gaussian noise sequences (stochastic spectra) were fed into a *linear* system, in this case a 6^{th} order low-pass Butterworth filter with a normalised cut-off frequency (at which point gain drops by 3 dB) of 0.5, where 1 corresponds to the Nyquist frequency. The filter coefficients were generated by the MATLAB command "`butter(6,0.5)`" and the sampling frequency was taken as unity. AWGN was added to the output such that the signal-to-noise ratio (SNR) was 20 dB. Within each of these experiments, M realisations of inputs, ranging from 1 (no averaging) to 64, were generated for each sub-experiments. Within each sub-experiment, the M outputs of both types of sequences were used to compute the transfer function estimates $\hat{G}(k)$ and the estimates were averaged across M using (4.5). For simplicity only $P = 1$ period of steady-state output was measured. The true transfer function $G_0(k)$ of the Butterworth filter was used to compute the errors. The 'average mean squared error (MSE)' is defined as the MSE across the M experiments, averaged across all frequency points, such that:

$$\text{Average MSE} \triangleq \frac{1}{MN} \sum_{k=0}^{N-1} \sum_{m=1}^{M} \left| \hat{G}^{[m]}(k) - G_0(k) \right|^2. \tag{4.10}$$

The average MSE was computed for each value of M, for each sub-experiment and for each input type; over the 500 experiments and plotted against M for both types

of input in Fig. 4.3a. The MATLAB code used for this experiment is available in Appendix B.3.

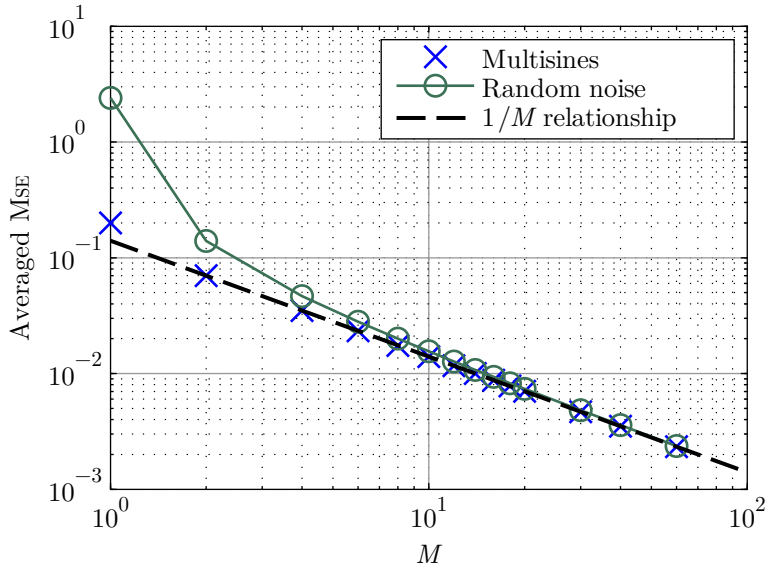
In addition, instead of continuous amplitude inputs, using discrete-level sequences of MLBS, Inverse-repeat Binary Sequence (IRBS) and DIRBS result in a similar picture in Fig. 4.3b.

The results show that, for sequences with static power spectra, the noise power is inversely proportional to $1/M$, i.e. the noise power decreases according to $\mathcal{O}(M^{-1})$ while for sequences with stochastic spectra, this is only asymptotically true. In some applications in industry, system identification requires halting production and the identification process may take a relatively long time due to the long settling time of some plant processes, especially chemical and thermal processes. The longer the experiment phase took, the higher the economic costs incurred. If the plant is nonlinear, the minimisation of cost contradicts the necessity to perform multiple experiments to obtain satisfactory results, due to the nature of nonlinear distortions as discussed in Section 4.1. This example illustrates that deterministic sequences with static power spectra are clearly advantageous due to their superior performance in low averaging amount. If a high (about $M > 10$) number of realisations are to be conducted, the performance of, for example, DIRBS and MLBS becomes very similar and the choice between them is less important. On the other hand, if only limited experiment time is available, deterministic sequences are recommended over stochastic ones. Pintelon and Schoukens (2012, sec. 2.6) provide a more detailed analysis on the topic of comparing sequences with stochastic and deterministic spectra.

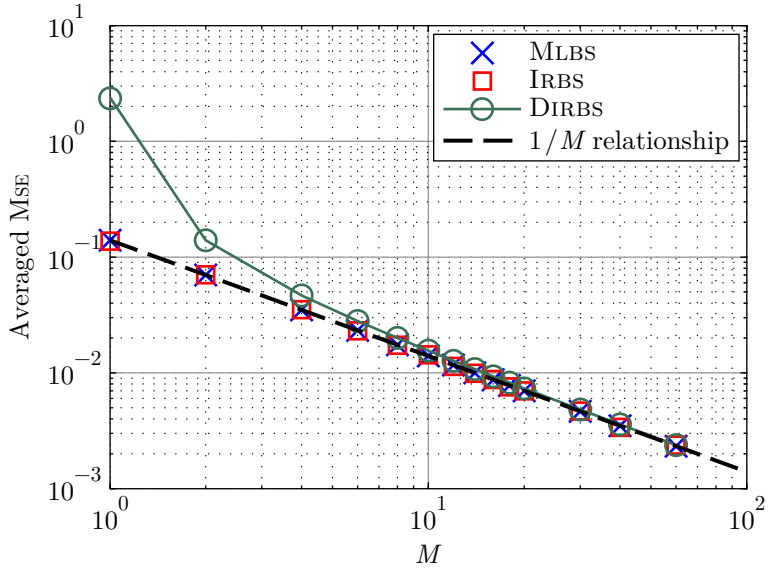
4.4 Use of binary sequences in obtaining the BLA

It is of interest whether binary sequences are viable inputs to obtain the BLA of a nonlinear system and how different types of binary sequences, such as the DIRBS, MLBS and IRBS, perform relative to each other. In this section, the three types of binary sequences are used to identify the BLAs of discrete-time Wiener systems with two different nonlinearities with and without disturbance noise added at the output. It will be seen that the choice depends greatly on the levels of even order nonlinear distortions. The work in this section was published in Wong et al. (2012c).

In Fig. 4.4, $u[k]$ is a periodic input of period N, $y[k]$ is the noise-free system output with the same period and $n[k]$ is the noise term and assumed to be aperiodic.



(a) Continuous sequences



(b) Discrete sequences

Figure 4.3: Performance comparison of various stochastic and invariant spectrum sequences against number of realisations of input

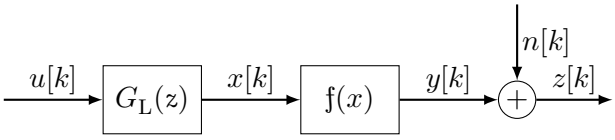


Figure 4.4: Discrete Wiener system block diagram with noisy output (OE model)

The input $u[k]$ and measured output $z[k]$ are related by:

$$z[k] = \mathfrak{f}\big(g[k] \circledast u[k]\big) + n[k]. \tag{4.11}$$

Similar to (4.1), a discrete-time equivalent description may be established:

$$z[k] = g_{\text{BLA}}[k] \circledast u[k] + \nu[k] + n[k] \tag{4.12}$$

where g_{BLA} is the BLA in time domain in the form of an impulse response function (IRF) and $\nu[k]$ is a periodic and unknown nonlinear distortion term due to deviations from the BLA.

4.4.1 Effect of even-order nonlinearities

Since the *odd* central moments of zero mean binary sequences (Table 3.1) are zero, the theoretical expression of the BLA as described by (3.24) would not be affected if there are *even* degrees of nonlinearities present. In practice, the contributions to the BLA from even order nonlinearities are only asymptotically ($M \to \infty$) zero (Pintelon & Schoukens, 2012, sec. 3.4.1).

As noted in Section 2.3.5, IRBS's are immune to even order nonlinear distortions. Using such sequences is expected to yield performance advantages in nonlinear systems with significant even order nonlinearities, in expense of experiment time relative to an MLBS due to the doubling in period.

Example Similar to the experiment in Section 4.3, another set of 500 independent simulation experiments was conducted. In this experiment, the performance of different periodic discrete-time binary sequences, namely, MLBS, IRBS and DIRBS, is compared in both noisy and noiseless scenarios. In the noisy case, exogenous AWGN sequences were added to the output and the SNR was set to 10 dB for all inputs. The linearity of the Wiener system was arbitrarily chosen to be a 10^{th} order window-based finite impulse response (FIR) low-pass filter with a normalised cut-off frequency of 0.5, generated by MATLAB command "`fir1(10,0.5)`" (IEEE Acoustics, Speech and Signal Processing Society. Digital Signal Processing Committee, 1979, Algorithm 5.2). Two types of nonlinearities were considered, firstly $\mathfrak{f}(x) = 0.5(x^2 + x^3)$, and secondly $\mathfrak{f}(x) = 0.5(x^3)$. The period N of the MLBS's and

DIRBS's were both chosen to be 8191, with that of the IRBS's equalled to 16 282. This choice ensured that the IRBS had the same spectral resolution as the other two input types. For IRBS's, only the excited frequency lines were considered, yielding the same frequency resolution as the other inputs.

Substituting $p_3 = 0.5$ into (3.24), the BLA in this case, applicable to both nonlinearities, is:

$$g_{\mathcal{U}_2\,\mathrm{BLA}}[r] = -a^4 g^3[r] + 1.5a^4 g[r] \sum_{k=0}^{\infty} g^2[k]. \tag{4.13}$$

This is used as the performance baseline for calculating the errors. The average MSE values are plotted against the average amount M, which is halved for IRBS only in order to impose fairness in experiment time requirement. The average MSE is defined similarly to (4.10), with the exceptions that firstly the average MSE is normalised by the Euclidean-norm of the transfer function of the FIR filter; and secondly the dc errors are ignored. The even order nonlinearities produce significant dc errors which IRBS's are immune to, hence including dc errors into the calculation of MSE would introduce considerable bias in favour of IRBS. The average MSE values for this example were evaluated using:

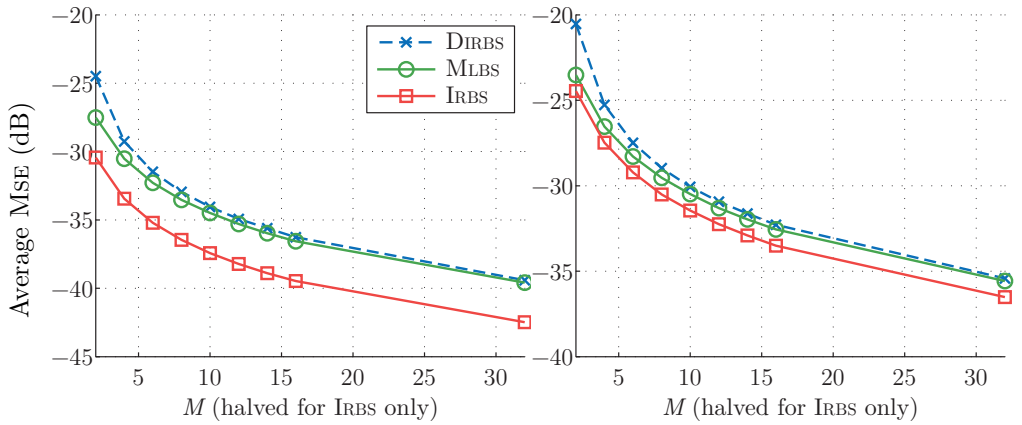
$$\text{Average MSE} = \frac{\frac{1}{M(N-1)} \sum_{k=1}^{N-1} \sum_{m=1}^{M} \left| \hat{G}^{[m]}[k] - G_{\mathrm{BLA}}[k] \right|^2}{\sqrt{\sum_{k=1}^{N-1} \left| G[k] \right|^2}} \tag{4.14}$$

The results are plotted in Fig. 4.5 and Fig. 4.6.

In the noisy cases, the MSE values are proportional to the noise power, and decrease according to $\mathcal{O}(M^{-1})$ as stated in Section 4.3.

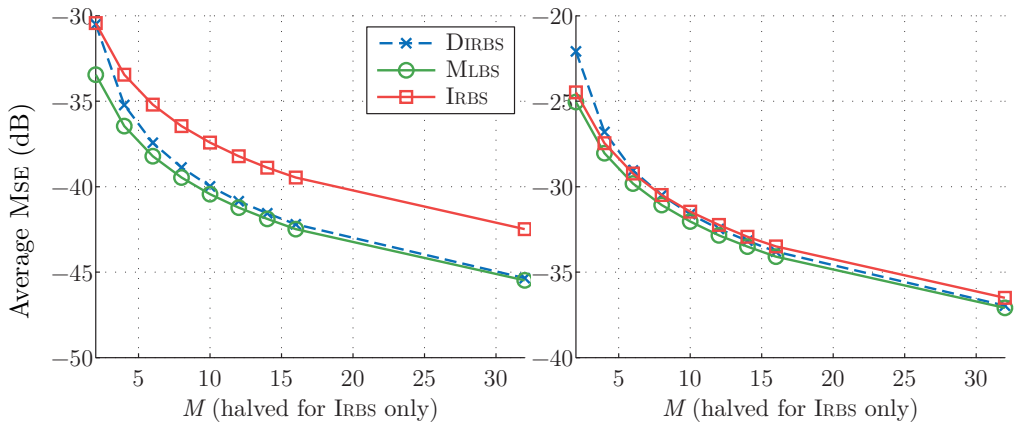
For the first set of experiments, with $\mathfrak{f}(x) = 0.5(x^2 + x^3)$, it can be seen from both Fig. 4.5a and 4.5b that the IRBS inputs achieve the best performance for all values of M. This is as expected since for these inputs, the contribution from the quadratic terms in the nonlinearity is zero. Of the other two inputs, MLBS proves better than DIRBS for low values of M, but the performance of these two classes of input becomes comparable as M increases. This is similar to the result for the linear case shown in Fig. 4.3.

For the second set of experiments, with $\mathfrak{f}(x) = 0.5x^3$, the IRBS's no longer have any advantages over their related MLBS's and this is confirmed by the results shown in Fig. 4.6. Recall the fairness criteria imposed on the experiment time such that M is halved for IRBS inputs. Because of this, IRBS's perform worse due to the loss of a factor of two in averaging in driving down the nonlinear distortions. This is



(a) Noiseless (b) 10 dB SNR

Figure 4.5: Performance comparison between IRBS, MLBS and DIRBS as inputs to a Wiener system with even and odd order nonlinear distortions



(a) Noiseless (b) 10 dB SNR

Figure 4.6: Performance comparison between IRBS, MLBS and DIRBS as inputs to a Wiener system with only odd order nonlinear distortions

confirmed by Fig. 4.6a that the performance of IRBS case when $M = 12$ on the plot is the same as the performance of MLBS when $M = 6$, with an asymptotic separation between two inputs of approximately 3.5 dB. In the noisy case of Fig. 4.6b, the picture is more complicated. With extraneous noise, only half of the frequency components were considered during identification when the input was IRBS and the frequency resolution was preserved. This means that the effective white extraneous noise power is also cut in half. Hence, the overall effect of using IRBS instead of MLBS in systems with only odd order nonlinear contributions in terms of tackling extraneous noise is nil; while the level of nonlinear distortion appears higher with IRBS due to less averaging when imposing fairness in experiment time. In Fig. 4.6 there is roughly a 8.5 dB increase in error power by introducing output noise for MLBS (from 4.6a to 4.6b). In comparison, the increase in the case of IRBS is only approximately $8.5 - 3.5 = 5$ dB.

Quick preliminary experiments can be performed to check the amount of even order nonlinearities present in a system before a decision is made whether to use an IRBS input—hence the sacrifice in frequency resolution or experiment time, is justified. For example, a short IRBS sequence may be used as an input to the system in question. If there is significant output power in even harmonic lines—an indication of presence of even order nonlinearities, then it may be worth considering using IRBS over MLBS.

Overall, the results show that the BLA can be measured using any of the three classes of binary sequences.

4.5 Structured behaviour of nonlinear distortions with MLBS inputs

In nonlinear system identification, results from traditional non-parametric model identification techniques contain both linear and nonlinear contributions. When Gaussian excitation signals (including random-phased multisines) are used, the nonlinear contributions are noise-like due to their asymptotic circular complex Gaussian distribution of the spectra (Pintelon & Schoukens, 2012, theorems 3.8 & 3.11). It is therefore difficult to distinguish nonlinear distortions from noise. However, unlike random noise, the nonlinear distortions are static in the sense that a particular input sequence will always produce the same nonlinear distortions; the distortions will only be different if another input sequence is used, even if the second input sequence is from the same class of sequences as the first. Hence, with the robust method outlined in the beginning of this chapter, the variances of the nonlinear distortion and

noise can be estimated separately by considering inter-period and inter-experiment variances.

For many nonlinear systems, if MLBS's are used as excitation signals, the resulting nonlinear distortions are highly structured due to the shift-and-multiply property of m-sequences (see p. 24). This has been illustrated for several systems with direction-dependent dynamics (i.e. those in which the dynamics are different when the output is increasing from those when the output is decreasing) (Rosenqvist, Tan, Godfrey & Karlström, 2006). It has also been illustrated on a full-scale industrial process—the *Pegasus Vertical Take-off Engine* (Godfrey & Moore, 1974)—in which the direction-dependent behaviour almost certainly arose from the fuel flow transducer being faster in the downward direction than that in the upward direction.

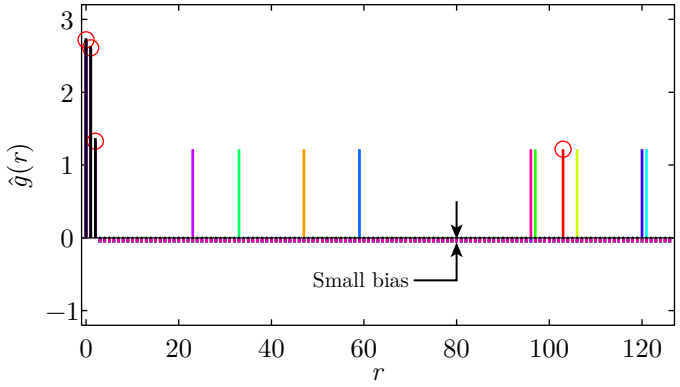
This section documents how the median averaging scheme may be advantageous over the conventional mean averaging scheme under structured nonlinear distortions. To enable fair comparison in terms of identification performance in terms of errors (see description of the theoretical BLA in Section 4.6), signals based on DIRBS (see Section 2.3.2), which do not possess the shift-and-multiply property, were also used. Part of the work in this section was published in Wong et al. (2013b).

To illustrate the difference between nonlinear distortions when using an MLBS input and that when using a DIRBS input, simulation experiments were performed on a Wiener system (Fig. 4.4) with a pure cubic nonlinearity. Both types of input have their power normalised to unity, and for each type, 10 experiments were performed without output noise such that $n[k] = 0\ \forall k$. Using (4.2) and (4.3) for each experiment independently (without averaging between experiments), ten IRF estimates of $\hat{g}[r] = \mathcal{F}^{-1}\{\hat{G}(\jmath\omega)\}$ were obtained, where $\mathcal{F}^{-1}$ represents the inverse DFT. The MLBS inputs were 10 different sequences, randomly selected from the 18 available MLBS's with $N = 127$; whereas the DIRBS's were 10 different segments of a DIRBS, all of length $N = 127$. The linear system was a 3-tap FIR filter with an IRF of $g_{\mathrm{L}}[k]$ and a frequency response function (FRF) of $G_{\mathrm{L}}(z) = \mathcal{F}\{g_{\mathrm{L}}[k]\}$ such that:

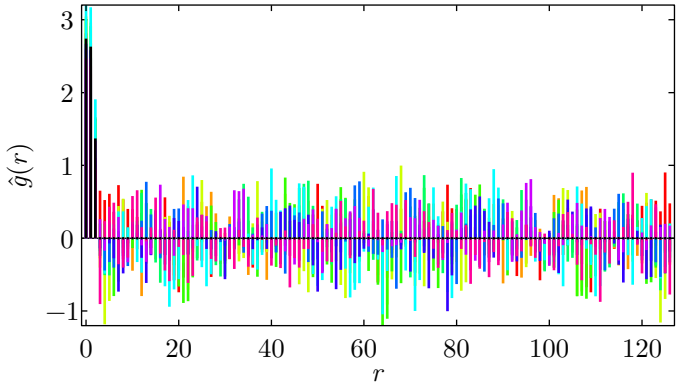
$$G_{\mathrm{L}}(z) = 1 + 0.7\,z^{-1} + 0.3\,z^{-2}\,. \tag{4.15}$$

The results are illustrated in Fig. 4.7, which has BLA estimates $\hat{g}_{\mathrm{BLA}}[r]$ obtained from 10 different MLBS's (Fig. 4.7a) and DIRBS's (Fig. 4.7b). It can be seen that the use of MLBS's results in nonlinear distortions appearing as isolated 'spikes' while for DIRBS's they appear more random. For the case of MLBS's and in the frequency domain, instead of spikes, the nonlinear distortions act over a wide frequency range with a complex structure.

To explain the difference between the behaviour of nonlinear distortions when



(a) MLBS



(b) DIRBS

Figure 4.7: Illustration of the difference in behaviour of nonlinear distortions given different inputs—(a) Structured combination for MLBS and (b) Random scattering for DIRBS. Different shades represent different experiments (10 in total), each with different input realisations, no averaging (i.e. $M = 1$).

using the two different types of inputs in the time domain, note firstly that:

$$y[k] = (u[k] + 0.7u[k-1] + 0.3u[k-2])^3 \tag{4.16}$$

and that the second order moment is unity for either class of signal with levels ± 1, i.e.:

$$u^2[k] = u^2[k-1] = u^2[k-2] = 1. \tag{4.17}$$

Applying the algebraic expansion of (4.16) and noting (4.17),

$$y[k] = 2.740u[k] + 2.632u[k-1] + 1.368u[k-2] + 1.26u[k]u[k-1]u[k-2]. \tag{4.18}$$

The theoretical expression for the BLA of a Wiener system with IRF of the linearity denoted by $g_{\mathrm{L}}[k]$ and a pure cubic nonlinearity, excited by *any* zero-mean

binary inputs, is given by (3.24) in Section 3.3.1. With power normalised to unity, (3.24) is simplified to:

$$g_{\mathrm{BLA}}[r] = 3g_{\mathrm{L}}[r] \sum_{k=0}^{\infty} g_{\mathrm{L}}^2[k] - 2g_{\mathrm{L}}^3[r]. \tag{4.19}$$

From (4.15), the time domain IRF ordinates are $g_{\mathrm{L}}[0] = 1$, $g_{\mathrm{L}}[1] = 0.7$,$g_{\mathrm{L}}[2] = 0.3$ and $g_{\mathrm{L}}[k \geq 3] = 0$. This gives $\sum_{k=0}^{\infty} g^2[k] = 1^2 + 0.7^2 + 0.3^2 = 1.58$. Substituting this value into (4.19) results in:

$$g_{\mathrm{BLA}}[r] = 4.74g_{\mathrm{L}}[r] - 2g_{\mathrm{L}}^3[r]. \tag{4.20}$$

With $r = 0, 1, 2, ...$, the BLA is therefore $g_{\mathrm{BLA}}[0] = 2.740, g_{\mathrm{BLA}}[1] = 2.632, g_{\mathrm{BLA}}[2] = 1.368$ and otherwise zero for $r \geq 3$. In the z-domain this is written as:

$$G_{\mathrm{BLA}}(z) = 2.740\, z + 2.632\, z^{-1} + 1.368\, z^{-2} \tag{4.21}$$

4.5.1 MLBS case

Reproducing the shift-and-multiply property described on p. 24:

$$u[k]u[k-a] = -u[k-\alpha] \quad \forall k \tag{4.22a}$$

$$u^2[k] \equiv 1 \quad \forall k \tag{4.22b}$$

where a and α are different non-zero integers. Following on from this,

$$u[k]u[k-a]u[k-b] = -u[k-\alpha]u[k-b] = +u[k-\beta] \tag{4.23}$$

where a, b and α are different non-zero integers and β is a non-zero integer $\neq \alpha$ and $\neq b$. This can be extended for higher order multiples of delayed versions of u. The values of a, b, α and β are uniquely dependent on the MLBS u.

From (4.23):

$$1.26u[k]u[k-1]u[k-2] = 1.26u[k-\gamma] \tag{4.24}$$

where γ is an integer depending on the particular MLBS so that (4.18) becomes:

$$y[k] = 2.740u[k] + 2.632u[k-1] + 1.368u[k-2] + 1.26u[k-\gamma]. \tag{4.25}$$

From (3.9), the estimated BLA is defined by $\hat{g}_{\mathrm{BLA}}[r] = \mathrm{E}\left[\!\left[y[k]u[k-r]\right]\!\right]$, so that it depends on the form of the autocorrelation function $R_{\mathrm{uu}}[k]$ of the sequence. By discretising (2.6), the autocorrelation function for an MLBS with levels ± 1 and of

period $N = 127$ is given by:

$$R_{\text{uu}}[k] = \begin{cases} 1 & k = 0 \\ -\frac{1}{127} & k \neq 0. \end{cases} \tag{4.26}$$

The four terms in (4.25) thus give rise to components of 2.740, 2.632, 1.368 and 1.26 at $r = 0, 1, 2$, and γ respectively, together with a small constant offset in each case due to the small non-zero value of the off-peak autocorrelation function. This small offset, visible in Fig. 4.7a, would be smaller if a longer MLBS were used. The values for the MLBS based on the primitive polynomial $\text{p}(x) = x^7 \oplus x^4 \oplus 1 = 0$, where $\oplus$ denotes modulo-2 addition (see, for example, Godfrey [1980]) are marked by circles in Fig. 4.7a; for this particular MLBS, $\gamma = 24 = N - 103$ (such that by periodicity, $u[t - 24] = u[t + 103]$, which corresponds to a spike at $r = +103$ in the figure). Except for the small offset, the first three coefficients are the same as the theoretical expression for the BLA for *any* zero-mean binary inputs given by (4.21).

To summarise, the $u[k-\gamma]$ term in (4.25) is manifested as 'spikes' in Fig. 4.7a, appearing at different locations specific to the particular realisation of the MLBS input used. Given a specific m-sequence, it may seem a good idea to predict where 'spikes' may occur and compensate for the nonlinear distortion. However, the computation of the location of spikes requires the explicit knowledge of both the linearity and nonlinearity that are unavailable, and the complexity grows with both the length of the impulse response and the degree of the nonlinearity. In addition, for more complicated nonlinearities with combined degrees such as cubic plus quintic, it is not feasible to separate the contribution of spikes in the output between the various degrees.

Another idea is to choose specific m-sequences to push the location of the spikes to where the impulse response is expected to be zero, usually near the end when the dynamics of the stable system has decayed to below the noise floor. Unfortunately there are only a limited number of MLBS's within a given family (or the same period, see Appendix A.1) and given a particular MLBS, one has no choice on where the spike occurs. The number of possible m-sequences increases as the period increases, so improving the chance that a favourable MLBS could be found, but as far as the author is aware, there are no tabulated values of delay resulting from the shift-and-multiply property for large values of period N. It may be possible, depending on experiment setup, to artificially use a long sampling time relative to the memory of the system, hence the spikes are more likely to be at the zero points and therefore more easily eliminated. Doing so, however, increases the bias of the BLA obtained (see Section 3.5.2).

4.5.2 DIRBS case

For a DIRBS input, (4.24) does not apply, and the triple product in (4.18) contributes nonlinear distortions to the estimated BLA, at all values of r. The phases of the DFT of a DIRBS are random (a property shared by random-phased multisines, see Sections 2.3.2 and 2.3.3) and because of this, the nonlinear distortions are indistinguishable in appearance from Gaussian output noise. For a DIRBS with levels ± 1, the autocorrelation function is, from (2.3), asymptotically equal to:

$$R_{\mathrm{uu}}[k] = \begin{cases} 1 & k = 0 \\ 0 & 0 < k \leq N-1 \end{cases} \tag{4.27}$$

as either the length N becomes very large or in the case of multiple segments, the number of segments M is such that the product MN is very large. Thus, for such a sequence, the first three coefficients would be those of the theoretical BLA, together with the nonlinear distortion term from the triple product; there would be no constant offset. However, when using finite length segments (in this case with $N = 127$), the autocorrelation will only be an approximation of (4.27), which adds a further random term to the estimated BLA. The estimated BLAS using 10 independent DIRBS segments are shown in Fig. 4.7b. As expected, there are now no additional spikes, and the nonlinear distortion has no noticeable pattern and could easily be taken as additive output noise.

4.5.3 Remarks on the system structure and the nonlinearity

While the Wiener system in this section has had quite a simple structure, the theory applies to systems with a more general structure such as that of a Volterra system. For example, the 3rd Volterra contribution is given by

$$y^{(3)}[k] = \sum_{a=0}^{N-1} \sum_{b=0}^{N-1} \sum_{c=0}^{N-1} h[a,b,c]u[k-a]u[k-b]u[k-c] \tag{4.28}$$

where $h[a,b,c]$ is a slice of the 3rd degree Volterra kernel. For the Wiener system with cubic nonlinearity considered in this section, only the 3rd Volterra kernel expansion has non-zero contributions; the Volterra kernel $h[a,b,c]$ is then equal to $g_{\mathrm{L}}[a]g_{\mathrm{L}}[b]g_{\mathrm{L}}[c]$ and (4.28) degenerates into (4.18) after simplification.

The theory illustrated so far has only considered static polynomial nonlinearities—in any finite interval, it is possible to achieve convergence, in least squares sense, of a polynomial to any static function as the degree of the polynomial tends to infinity. Although the theoretical analysis is beyond the scope of the thesis, struc-

tured nonlinear distortions were also observed in simulation experiments involving MLBS inputs with other classes of nonlinearities such as sinusoidal and non-integer power functions (e.g. square roots). Nonlinearities having hysteresis and memory effects have not yet been investigated, but Godfrey and Moore (1974) have shown that even in such cases, the nonlinear distortion can be structured.

4.5.4 Merits and demerits of the median estimator

As noted in Section 4.2, the conventional way of reducing the effect of nonlinear distortions when estimating a BLA is to use several different realisations of the input u and then to average them. Conventionally, averaging is performed using the arithmetic mean. For an MLBS input, multiple unique realisations of m-sequences would be used. However, knowing that for m-sequence inputs the nonlinear noise is 'spiky', i.e. outliers are expected, it is interesting to examine whether an alternative method of averaging, such as the median, can be used in this case to improve convergence compared with the mean. To generalise the averaging operation, (4.5) is modified to:

$$\hat{G}(\mathrm{j}\omega) \approx \frac{\hat{S}_{\mathrm{ZU}}(\mathrm{j}\omega)}{\hat{S}_{\mathrm{UU}}(\omega)} \triangleq \frac{\underset{\substack{p:1\to P\\ m:1\to M}}{\mathrm{Avg}} S_{\mathrm{ZU}}^{[m,p]}}{\underset{\substack{p:1\to P\\ m:1\to M}}{\mathrm{Avg}} S_{\mathrm{UU}}^{[m,p]}} = \frac{\underset{\substack{p:1\to P\\ m:1\to M}}{\mathrm{Avg}} \left(Z^{[m,p]}\overline{U}^{[m,p]}\right)}{\underset{\substack{p:1\to P\\ m:1\to M}}{\mathrm{Avg}} \left(U^{[m,p]}\overline{U}^{[m,p]}\right)}. \tag{4.29}$$

Relative Statistical Efficiency (RSE)

In statistics, the term *central tendency* (location) describes the fact that quantitative data tend to cluster around some value and this tendency is quantified through location estimators such as, but not limited to: mean, mode and median (Dodge et al., 2006). The expected value of a small sample M drawn from a Gaussian parent population follows a Student's t-distribution with $\nu = M-1$, where ν is the number of degrees of freedom. Estimating the central tendency with the mean, median or other location estimators yield results with different uncertainties or variances associated (Kenney & Keeping, 1962).

Definition If the mean estimator is used as a baseline, the ratio of uncertainties of a location estimator relative to the baseline is called the relative statistical efficiency (RSE), i.e.

$$\mathrm{RSE} = \frac{\mathrm{E}[\![\sigma^2_{\mathrm{LE}}]\!]}{\mathrm{E}[\![\sigma^2_{\mathrm{mean}}]\!]} \tag{4.30}$$

in which σ^2_{LE} and σ^2_{mean} are the variances associated with the central tendency estimated using either an arbitrary location estimator (LE) or the arithmetic mean, respectively, of a population when sampling with a given sample size M. The expectation operators E operate across an infinite number of independent experiments in drawing samples from the same population.

A Monte Carlo simulation with more than ten million trials was conducted to obtain the RSE of the median estimator with regards to the mean estimator for M between 2 and 16 inclusive. In each trial and for each M value, M random data were generated using a zero-mean Gaussian process with variance of unity. The mean and median of the vector of M values was separately calculated and recorded for the ten million trials. The ratio of the variances between the median values to the mean values gives an estimate of the RSE. The result of the experiment is shown in Fig. 4.8. Approximate values of RSE under the same context are also tabulated in (Kenney & Keeping, 1962) and are in agreement with those obtained from the simulation experiment.

Note that in the literature, the asymptotic relative (statistical) efficiency (ARE) is often quoted (Serfling, 2011). The term 'asymptotic' refers to the convergence behaviour for large sample size, and is different to the RSE used in the later experiment in Section 4.6. This is because statistical efficiency depends on the variance of sampling distributions (the distribution after sampling a parent distribution) of the location estimators involved; valid expressions of which may only be available asymptotically. The sampling distribution for, say, the median estimator is approximately normal with variance $\left[4f^2M\right]^{-1}$ where f is the population probability density function (p.d.f.) (Serfling, 2011). However, this is an approximation and valid only for a large sample size, whereas in this experiment one is interested in the behaviour of the statistical efficiency of various estimators with a small sample size. Otherwise, the underlying distribution will become Gaussian as Student's t-distribution also tends to Gaussian and the *asymptotic* relative efficiency then applies. Hence, the ARE of a Student's t-distribution is, in essence, the ARE of a Gaussian distribution. This is why, instead of the more commonly quoted ARE, the RSE is used to explain the difference between the two estimators for this particular setup.

One can observe there is a zigzag trend in Fig. 4.8—a dependence on whether the sample size is even or odd. This is because when M is even, the median is defined as the arithmetic mean of two 'middle' values. This is also why, for a sample size of two, the RSE is exactly one—for mean and median are essentially the same

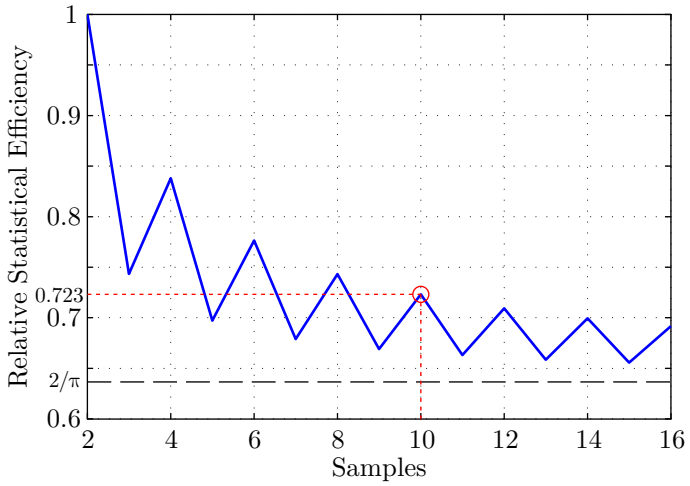


Figure 4.8: RSE of the median and the mean against sample sizes in estimating Gaussian population mean.

mathematical operation. It is interesting to see that even sample sizes result in a higher efficiency for the median estimator.

Two values of interest here are: 0.723 for $M = 10$ (the sample size used in the example in this chapter), which is in agreement with the approximate value of 0.71 given by (Kenney & Keeping, 1962); and $2/\pi$, which is the ARE as $M \to \infty$ as noted in Serfling (2011).

4.5.5 The median estimator

The median and the mean estimators both have the same objective: estimating the central tendency of a parent population given limited samples drawn from it. The median estimator is more robust in the sense that it is more resilient against influence of extreme values compared to the mean function (Kenney & Keeping, 1962, p. 54).

When applied to estimation of the BLA, one is relying on the fact that the dominant impulse response ordinates of the BLA are above the noise floor and hence the 'spikes' arising from structured combination of these impulse response ordinates are sufficiently distinctive from the output noise, as described in Section 4.5.1. If this is true, the median would be expected to perform better than the mean. However, if the opposite is true, i.e. the noise level is higher than that of the nonlinear distortions, the performance of this method would be slightly worse than conventional mean-based averaging method due to inherent inferior RSE with the median estimator when the output noise usually has, in practice, a Gaussian distribution. As such,

one expects the performance of the median estimator to depend on the level of the output noise relative to that of the nonlinear distortions.

4.5.6 Hodges-Lehmann Location Estimator (HLLE)

When the SNR for a system is moderate and one is unsure about the relative contributions of the nonlinear distortions to environment noise, it may be advantageous to use an averaging scheme that offers a good compromise between statistical efficiency and robustness to outliers. The Hodges-Lehmann location estimator (HLLE) is one such scheme (Serfling, 2011). The HLLE is a consistent and unbiased estimator of the population median. For data set $x \in X$ with n members the HLLE is the median of pairwise means such that $\overline{x}_{\text{HL}} = \text{Med}\big[\big\{\frac{1}{2}(x_i + x_j)\big\}\forall 1 \leq i < j \leq n\big]$.

The ARE of the HLLE with respect to the mean estimator is $^3/_\pi = 0.955$ to 3 significant figures (Serfling, 2011).

4.6 Experimental comparison between the mean and median averaging

To gauge how much performance one may expect to gain or lose from replacing the mean by the median, results are presented from simulation experiments, in which both the ideal scenario without output noise and scenarios with various levels of noise were considered. Two sets of the classes of inputs, both with period $N = 8191$ were used to excite a discrete-time Wiener system with structure shown in Fig. 4.4; the first type being m-sequences with levels ± 1 described in Section 2.3.4, and the second type periodic DIRBS's with zero expected mean as described in Section 2.3.2. The linearity was chosen to be a 10^{th} order infinite impulse response (IIR) Chebyshev Type-I (equiripple in pass-band) band-pass filter with an IRF of $g_{\text{L}}[k]$ and with a pass-band normalised edge frequencies of 0.2 and 0.6, where a normalised frequency of 1 corresponds to the Nyquist frequency. The Chebyshev filter was generated via the MATLAB command "`cheby1(10,0.5,[0.2,0.6])`". It is important to note that this IIR filter has a relatively long settling time compared to, say, the 10^{th} order FIR filter used in Section 4.4.1. A linearity with a long memory represents a sub-optimal scenario in terms of performance gain of median over mean averaging, because the nonlinear distortions will be more spread out and less pronounced. This will be discussed at the end of the subheading '*Comparison without output noise*' in Section 4.6.1. Identical to the second experiment setup in Section 4.4.1, the static nonlinearity was of the form $\mathfrak{f}(x) = 0.5(x^3 + x^2)$.

With reference to (4.29), for simplicity no inter-period averaging was performed so that $P = 1$. The measurements were taken after steady-state was reached such that transient effects were either nil or negligible. The number of realisations M was chosen to be 10 so that the performance between DIRBS and MLBS due to stochastic and deterministic power spectra is similar (see Fig. 4.3).

Unlike the mean function, the median function is not a linear operation; differences arise depending on whether the averaging takes place before or after the application of inverse DFT. Due to the lack of closed-form solution for using median in averaging complex numbers, i.e. the complex terms in the cross-power spectra at the numerator of (4.29), averaging is best performed in the time domain. In addition, there are no advantages in averaging the auto-power spectra (the denominator of 4.29) using any schemes other than the mean function due to its known Gaussian nature. For this simulation experiment, with $P = 1$, (4.29) is modified to:

$$\hat{g}_{\text{BLA}}[r] = \underset{m:1\to 10}{\text{Avg}}\ \mathcal{F}^{-1}\left\{\frac{S_{\text{ZU}}^{[m]}}{\frac{1}{10}\sum_{m=1}^{M} S_{\text{UU}}^{[m]}}\right\}. \tag{4.31}$$

The function Avg in (4.31) is defined as either *mean*, *median* or *HLLE* and their performance is compared for both MLBS and DIRBS inputs.

To compare the performance of various experiment regimes, a useful reference to use for calculating errors is the theoretical 'true' transfer function or impulse response; this is the theoretical BLA of the system and is dependent on, among various factors, the input amplitude distribution, e.g. binary or Gaussian. For a Wiener system consisted of a linearity with an IRF of $g_{\text{L}}[k]$ followed by the nonlinearity of $\mathfrak{f}(x) = 0.5(x^3 + x^2)$, given any zero-mean periodic binary inputs of length N with amplitude levels normalised to unity, the expected BLA is reproduced from (4.13) in Section 4.4.1 as:

$$g_{\text{BLA}}[r] = 1.5 g_{\text{L}}[r] \sum_{k=0}^{N-1} g_{\text{L}}^2[k] - g_{\text{L}}^3[r]. \tag{4.32}$$

This theoretical BLA will form the baseline reference so that the MSE used in subsequent calculations is defined as:

$$\text{MSE} = \frac{1}{N}\sum_{k=0}^{N-1} \left|\hat{g}[k] - g_{\text{BLA}}[k]\right|^2. \tag{4.33}$$

4.6.1 Results and analysis

For the experiment setup outlined in Section 4.6, two scenarios were considered; firstly without output noise disturbances and secondly where AWGN sequence $n[k]$

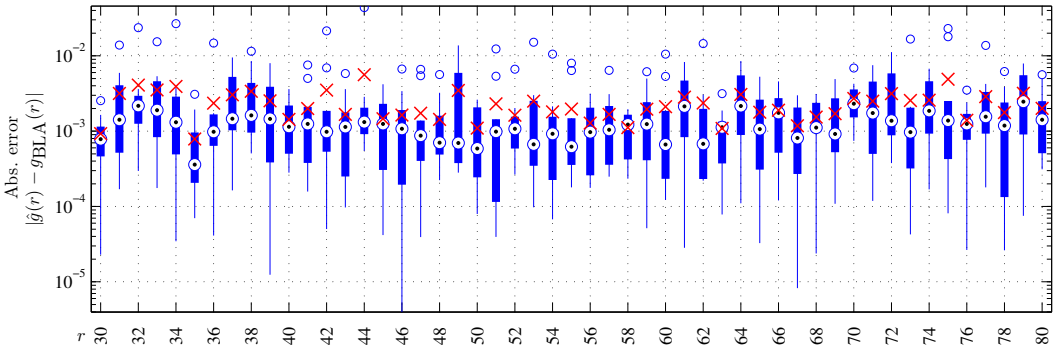


Figure 4.9: Log scale box-plot of absolute estimation errors of a section of a best linear impulse response $\hat{g}[r]$ for r between 30 and 80 using $M = 10$ different M-sequences without output noise disturbances.
Boxes: Interquartile range (IQR), *Whiskers*: 1.5×IQR of the upper or lower quartiles, *Circles*: Outliers, *Circles with central dots*: Median, *Crosses*: Mean.

was added to the output of the nonlinearity $y[k]$ to give the measured output $z[k]$. In the second scenario, the power of the additive noise $n[k]$ was varied for each experiment so that a range of SNRs between -30 dB and 60 dB was obtained.

Comparisons with no output noise

Fig. 4.9 is a box-plot which illustrates how the median function (circles with central dot) can perform better than mean function (crosses) as the averaging tool when outliers are expected. It shows the distribution of the resulting absolute errors from estimating a part of the BLA of a system described in Section 4.6 compared to the theory in (4.32). Note that the y-axis is plotted on a logarithmic scale. Looking at $r = 34$ and 44 for example, the presence of extreme outliers (small open circles) affects the quality of the mean estimate significantly while the median estimate is not disturbed as much. While the median estimates may not always be better than the mean estimates (especially when noise is present), overall one expects the median estimates to be better at low noise levels.

Fig. 4.10 depicts the averaging performance of the mean and median estimators for different amount of averaging (as a function of M) when inputs were MLBS's. Even with just $M = 3$ data sets being averaged, the median estimator offers a solid advantage due to the behaviour of the nonlinear distortions described in Section 4.5.1. Note that when $M = 2$, the median and mean are mathematically the same operation, hence the identical performance. There is a slight zigzagging trend observed on the median results, this is not an artefact of spurious origin and

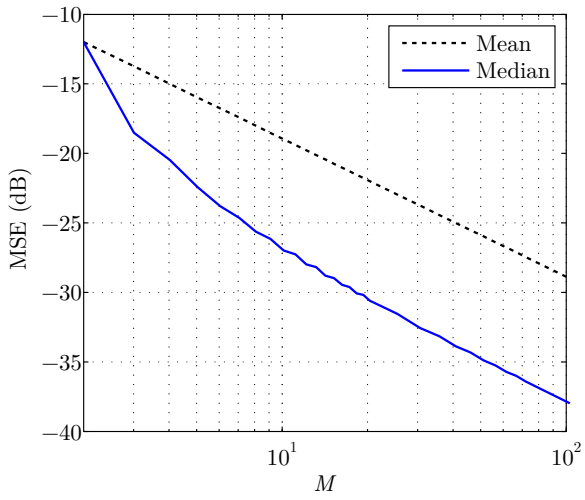
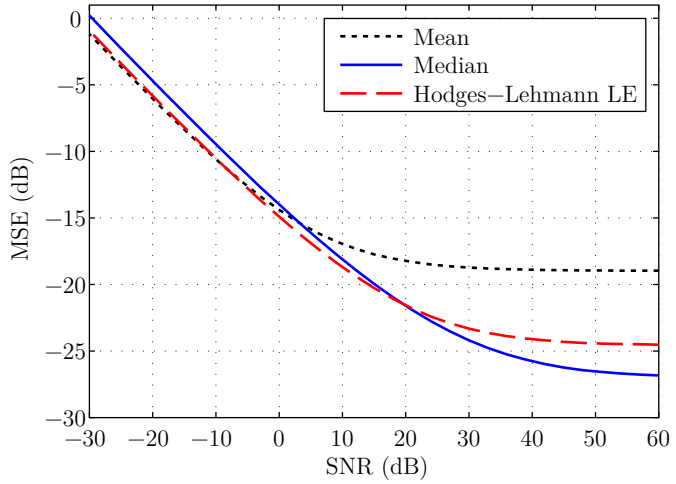


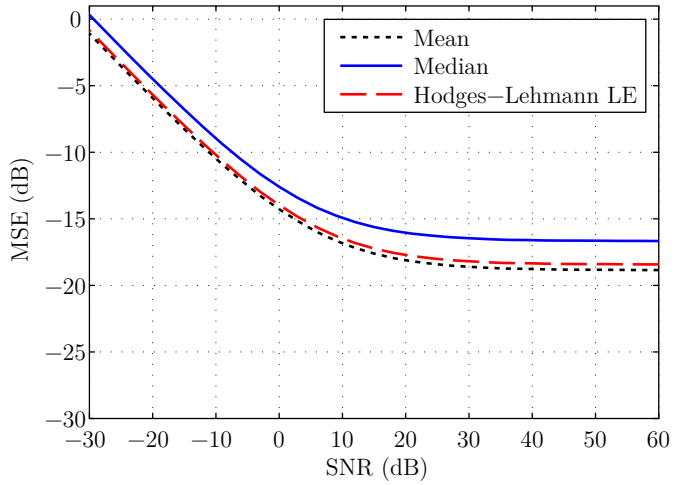
Figure 4.10: Estimation error of the BLA from output-noise-free measurements against number of MLBS input realisations (M) used in averaging when using either mean or median averaging schemes.

was explained in Section 4.5.4 through Fig. 4.8. The MSE from the mean estimator on the other hand follows a simple $\mathcal{O}(M^{-1})$ relationship as stated in Section 4.3.

Generally speaking, for systems with shorter memory (i.e. where the length of the IRF is short in terms of the sampling interval), the 'spikes' become less frequent but more pronounced. To explain this, consider again (4.25); with a non-zero IRF containing only 3 ordinates, only one spike can form for any given MLBS. If there are 4 non-zero ordinates, spikes will form on terms involving, for instance, $u[t]u[t-1]u[t-2]$, $u[t]u[t-1]u[t-3]$, $u[t]u[t-2]u[t-3]$ and other combinations; a system with a long IRF on the other hand, has its 'energy content' spreading across numerous ordinates, resulting in spikes that are less prominent. This is the case for this setup as the linearity used was an IIR filter with a long settling time. One can see even in such scenario, the median estimator can offer a substantial advantage. If the length of the system memory is short, depending on the levels of the environment and measurement noise, the median scheme can almost exactly reproduce the BLA of the system (even with limited averaging) due to its ability to completely decouple the estimated IRF from the influence of the dominating nonlinear distortions (in form of spikes). This allows the median function to gain a significant advantage over the mean function. The converse is also true: if the system memory is long, the advantage of the median compared to the mean is diminished due to the increased in frequency of appearance of spikes with lower magnitudes arising from the more spread out nature of the IRF—as the magnitudes are less extreme, the mean function



(a) MLBS



(b) DIRBS

Figure 4.11: Identification performance of various averaging schemes against SNR levels with (a) MLBS and (b) DIRBS inputs. Averaged results from 500 experiments each with $M = 10$.

is less affected by bias.

Comparison with output noise

Fig. 4.11 shows the sensitivity of different averaging methods to different noise levels (SNR). The experiment was repeated 500 times independently and the resulting MSE values were averaged. The calculation of SNR did not factor in the nonlinear distortions and hence a higher SNR can be thought of as having a higher contribution of nonlinear noise relative to the output noise.

Looking at the MLBS case in Fig. 4.11a, the median estimator has a significant advantage over the typical mean estimator for SNR above approximately 3 dB, due to the character of nonlinear distortions involved being 'spiky' in nature as predicted in Section 4.5.4. Below 3 dB, the mean estimator retains an almost constant advantage over the median estimator. As noted before, this is due to the lower statistical efficiency of the median estimator when the underlying dominating noise contribution is predominantly Gaussian. At -30 dB SNR the separation of MSE between the mean and median estimators is about 1.3 dB, which translates to a factor of 0.725. This is in very good agreement with the simulated RSE of 0.723 quoted in Section 4.5.4 for $M = 10$ in which the underlying population distribution was Gaussian.

For comparison, Fig. 4.11b shows the performance of the two averaging schemes when the inputs were periodic DIRBS's. In this case the combined nonlinear distortion and output noise contribution has a Gaussian distribution (see Sections 2.3.2 and 4.5.2). One expects the mean estimator to perform better overall due to the fact that the mean function is the maximum likelihood estimator (MLE) for variables with a Gaussian distribution (Pintelon & Schoukens, 2012, p. 24). The performance of which is also very close to the HLLE—the ARE of HLLE (relative to mean estimator) is 0.955; this is indeed the case. The median results are separated from the mean results by a ratio equal to the RSE, which is close to the 0.723 value noted before. The separation is also relatively constant and therefore independent of the SNR, unlike the case of m-sequence inputs.

It does not appear possible to make a theoretical analysis of which of median and mean averaging is the better given an arbitrary SNR value, unless the linearity and the nonlinearity are known; in this exercise, it was assumed that they are unknown. Note that both the mean and median estimators are computationally inexpensive to apply, so that the author recommends using both and checking the resulting models against the validation data if possible. In any case, assuming exogenous noise was Gaussian and dominating, the lower statistical efficiency of median would only translate into a slight increase in the MSE (1.3 dB from the RSE for $M = 10$). In contrast, a substantial gain may be obtained if the nonlinear distortions were dominating, especially if the underlying linearity had a short IRF where 'spikes' would be fewer and more prominent (see the last paragraph of the previous subheading).

Lastly, it is possible to combine the optimal performance of the median estimator in averaging out AWGN and the effectiveness of the median in driving out the structured nonlinear distortions. With the robust method described in Section 4.2,

it is possible to perform mean averaging independently over P, where nonlinear distortions are periodic and environment noise is usually aperiodic. With sufficient P for each realisation of m, it is possible to obtain M high quality estimates of the BLA compounded with nonlinear distortion terms. The median estimator can then be used to average over M to effectively drive down the nonlinear distortions.

4.7 Conclusions

Sections 4.1 and 4.2 have introduced a robust method for estimating the BLA of a nonlinear system, which relies on averaging estimates from multiple measurement records. The example in Section 4.3 has illustrated that deterministic sequences with static power spectra are clearly advantageous due to their superior performance in low averaging amount. For sequences with static power spectra, the noise power is inversely proportional to $1/M$ while for sequences with stochastic spectra, this is only asymptotically true. While so far the method and general result are known in the literature, the sections serve as a verification for the applicability to binary inputs.

The experiment in Section 4.4 compares the performance of various periodic binary sequences, namely DIRBS's, MLBS's and IRBS's in terms of averaging performance in sequestering both measurement noise and nonlinear distortions. The IRBS's are useful when the system has significant even order nonlinearities. On the other hand, using IRBS's requires twice the experiment time due to their period being twice that of their MLBS's counterparts. Supporting the findings in Section 4.3, DIRBS's are not as good as MLBS's when the number of averages is low.

Section 4.5 introduced the unique structured behaviour of the nonlinear distortions when the excitation signal used is an MLBS, resulting in 'spikes' forming in the estimated IRFs of the nonlinear systems. With respect to this behaviour, the subsequent sections discuss the potential gain when using MLBS's in combination of the median averaging scheme with reasonably high SNR. The combined use of MLBS's as excitation signals and the use of median averaging scheme was found to be better than the typical mean based estimator. Depending on the system memory and noise levels, in some cases the median estimator can completely eliminate the nonlinear distortions with limited averaging (see p. 80). In the setup considered, the use of robust averaging schemes gives 5 dB to 8 dB improvement in MSE performance, compared to using traditional mean based averaging. Since noise power decreases according to $\mathcal{O}(M^{-1})$, doubling the amount of available data reduces the

noise variance by a factor of two (3.03 dB); hence 5 dB to 8 dB translates to a reduction by a factor of approximately 3–6 in experiment time in data acquisition.

The number of 'spikes' is determined by the degree of the nonlinearity and the memory of the impulse response function $g_{\mathrm{L}}[k]$. For $g_{\mathrm{L}}[k]$ with a long settling time, such as an IIR filter, as in the examples in Sections 4.6 and 4.6.1, it is not immediately clear looking at the result whether MLBS's or DIRBS's were used, because the spikes are so numerous that the nonlinear contributions seem indistinguishable from Gaussian noise. Note that the proposed scheme takes advantage of distribution of the nonlinear distortions between different input realisations, and this is not to be confused with the appearance of nonlinear distortions as a function of time.

CHAPTER 5

Design of Multilevel Signals for Gaussianity

With respect to the identification of systems in practice, maximum-length sequences are often used, because of the ease of generation using, for example, the programs *prs* for binary sequences and Galois for pseudorandom multilevel sequences (Prms's) (Godfrey, Tan et al., 2005). Pseudorandom sequences are advantageous in situations where practical constraints limit the number of sequence levels that can be applied. Pseudorandom binary sequence (Prbs) have been widely used for many years, but there are also instances where multilevel sequences are necessary. In an example from the steel industry (Barker & Godfrey, 1999), ternary sequences were used to identify the frequency response between the applied force and steel strip position on a scale model of a hot-dip galvanising process. In this application, the strip could be moved by two electromagnets, one on either side. However, the associated power electronics only allowed one voltage level to be applied to each, thus limiting the input sequence to a maximum of three levels. An example from the field of communications is the identification of fibre-based wireless systems, where a simpler transmitter structure can be used if the input sequences are either binary or ternary (Ng, Tan & Chuah, 2011). A further example is the identification of an electronic nose described in Tan and Godfrey (2004). In this system, there were four compartments, which could be filled with different chemicals or the same chemical but with different concentrations. A metal oxide semiconductor sensor was exposed to the content in one of the compartments at any particular time and the input was implemented using four on-off valves. In Tan and Godfrey (2004), only two compartments were used but the physical structure of the system

allows a maximum of four input levels to be applied.

As noted in Chapter 3, the Best Linear Approximation (BLA) depends on the power spectra and the amplitude distributions of the inputs $\{\langle u \rangle\}$. The BLA when using a Gaussian input sequence has been well studied (Pintelon & Schoukens, 2012; Schoukens, Lataire et al., 2009). When non-Gaussian inputs are used, the BLA obtained is biased with respect to the Gaussian case. The amount of bias depends on both the form of the nonlinearities and the higher order moments of the input sequence. This bias, called the Discrepancy Factor (DF), was discussed in detail in Section 3.5 and is summarised in Section 5.1. DF was initially conceived as an *empirical* data-driven measurement metric to quantify the bias (see Section 5.1), *theoretical* expressions for which have been derived for the case of a discrete-time Wiener-Hammerstein (WH) system in Section 3.3 with a cubic or a quintic nonlinearity when excited by a white input with an arbitrary probability mass function (p.m.f.).

This chapter shows that it is possible to design discrete multilevel sequences to mimic Gaussianity as closely as possible (hence reducing the bias) by adjusting sequence levels and the probabilities of the sequence being at these levels. Their performance has been compared with the Gaussian case in simulation experiments. In particular, multilevel sequences are designed in this chapter with moments (see Section 3.2) as close as possible to those of a zero-mean Gaussian sequence $\langle u \rangle \sim \mathcal{N}(0, \sigma^2)$, for which the m^{th} moment $M_{\text{m}} \equiv \mathrm{E}[\![u^m]\!] = (m-1)!!$ for even m and zero for odd m. Here, $(m-1)!!$ is the double factorial of $(m-1)$ given by $(m-1)(m-3)(m-5)\ldots 1$. The closer the moments of an arbitrary sequence match the moments of $\mathcal{N}(0, \sigma^2)$, the closer the estimated BLA will be to that estimated using a Gaussian input sequence.

Tuning the higher order moments of a random input to match those of a Gaussian input is not the only method to design a random input for Gaussianity. For example, one can subdue the difference between the probability mass function (p.m.f.) of a discrete-level random input and a Gaussian probability density function (p.d.f.) through measures such as the Kullback-Leibler divergence (Kullback, 1987) or the Bregman divergence (Frigyik, Srivastava & Gupta, 2008). However, as far as the reduction of DF is concerned, when the exact details of the nonlinearity are unknown, moment matching is a simple and effective method. This is because the DF is shown to be directly dependent on the higher order moments of the inputs in Chapter 3. The effectiveness of this approach is demonstrated by the later simulation experiments in Sections 5.3.1 and 5.3.2.

5.1 Discrepancy Factor

It is useful to be able to quantify the difference between BLAs estimated using Gaussian input sequences and those estimated using non-Gaussian input sequences (in both cases, with power normalised to unity), and a convenient way of doing this is to use the empirical DF $\mathfrak{D}$ defined by Section 3.5 as:

$$\mathfrak{D} \triangleq \frac{\sum_{k=0}^{N-1} |\hat{g}_{\text{BLA Gaussian}}(k) - \hat{g}_{\text{BLA non-Gaussian}}(k)|^2}{\sum_{k=0}^{N-1} |\hat{g}_{\text{BLA Gaussian}}(k)|^2} \tag{5.1}$$

where N is the period of the input sequence (assumed to be longer than the settling time of the system) and $\hat{g}_{\text{BLA Gaussian}}(k)$ and $\hat{g}_{\text{BLA non-Gaussian}}(k)$ are the Gaussian and non-Gaussian BLA estimates, respectively.

5.2 Designing multilevel sequences to minimise the Discrepancy Factor

The DF can be driven down by making as many moments $\mathfrak{M}_n$ of a multilevel sequence as possible equal those of a Gaussian sequence with the same power. This will be developed here for ternary, quaternary and quinary sequences, all with levels symmetrical around zero and with zero mean. This results in all odd moments being zero, so that it is possible to concentrate on matching the even moments. The power of the sequences will be normalised to unity, which results in the moment $\mathfrak{M}_2$ being matched in each case. Due to the increasing degrees of freedom offered by the increasing number of amplitude levels and their associated probability variables, it is then possible to match one further moment for a ternary sequence, two for a quaternary sequence, three for a quinary sequence and so on. This is tabulated in Table 5.1 and the subsequent subsections show the design process involving the amplitude levels and their associated probabilities for the three symmetric sequences.

Table 5.1: Example matching of moments for Gaussianity for discrete symmetric sequences with various number of levels

Levels	**D.o.f.**[†]	**Matched moments**
2	1	2^{nd} (power)
3	2	2^{nd}, 4^{th}
4	3	2^{nd}, 4^{th}, 6^{th}
⋮	⋮	⋮

[†]Degrees of freedom

5.2.1 Ternary sequences

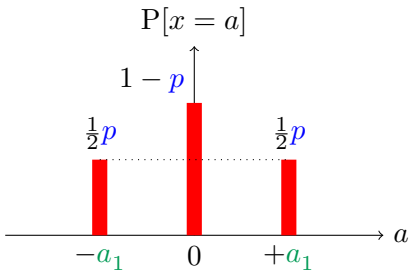


Figure 5.1: Designing the p.m.f. of a discrete symmetric ternary sequence

Referring to Fig. 5.1, let the levels be $\pm a_1$ (with equal probability $0.5p$) and zero, with probability $(1-p)$. This gives 2 degrees of freedom in the form of a_1 and p; for example, normalising the power ($\mathfrak{M}_2$) to unity and matching $\mathfrak{M}_4$ with that of a Gaussian sequence gives:

$$\mathfrak{M}_2 \triangleq \mathrm{E}\left[\!\left[u^2\right]\!\right] = a_1^2 p = 1 \tag{5.2a}$$

$$\mathfrak{M}_4 \triangleq \mathrm{E}\left[\!\left[u^4\right]\!\right] = a_1^4 p = 3!! = 3. \tag{5.2b}$$

Solving these for a_1 and p gives:

$$a_1 = \sqrt{3} \quad \text{and} \quad p = \frac{1}{3} \tag{5.3}$$

so that the sequence levels are ± 1.732, each with probability 0.1667, and zero, with probability 0.6667, to four significant figures. The p.m.f. of this tuned sequence is plotted in Fig. 5.2.

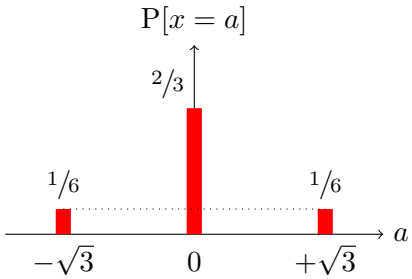


Figure 5.2: The p.m.f. of a symmetric ternary sequence tuned for Gaussianity

5.2.2 Quaternary sequences

Referring to Fig. 5.3, let the levels be $\pm a_1$, with equal probability $0.5p$, and $\pm a_2$, with equal probability $0.5(1-p)$. This gives 3 degrees of freedom in the form of

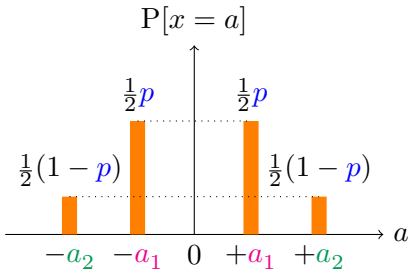


Figure 5.3: Designing the p.m.f. of a discrete symmetric quaternary sequence

a_1, a_2 and p. Normalising the power to unity and matching $\mathfrak{M}_4$ and $\mathfrak{M}_6$ with those of a Gaussian sequence gives:

$$\mathfrak{M}_2 \triangleq \mathrm{E}\left[\!\left[u^2\right]\!\right] = a_1^2 p + a_2^2(1-p) = 1 \tag{5.4a}$$

$$\mathfrak{M}_4 \triangleq \mathrm{E}\left[\!\left[u^4\right]\!\right] = a_1^4 p + a_2^4(1-p) = 3!! = 3 \tag{5.4b}$$

$$\mathfrak{M}_6 \triangleq \mathrm{E}\left[\!\left[u^6\right]\!\right] = a_1^6 p + a_2^6(1-p) = 5!! = 15. \tag{5.4c}$$

Solving these gives:

$$a_1 = \sqrt{3-\sqrt{6}}, \quad a_2 = \sqrt{3+\sqrt{6}},$$
$$p = \frac{1}{6}\left(3+\sqrt{6}\right). \tag{5.5}$$

so that the sequence levels are ± 0.7420, each with probability 0.4541, and ± 2.334, each with probability 0.0459, to four significant figures. The p.m.f. of this tuned sequence is plotted in Fig. 5.4.

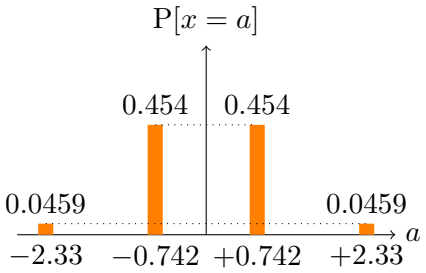


Figure 5.4: The p.m.f. of a symmetric quaternary sequence tuned for Gaussianity

5.2.3 Quinary sequences

Referring to Fig. 5.5, let the levels be $\pm a_1$, with equal probability $0.5p_1$; $\pm a_2$, with equal probability $0.5p_2$; and zero, with probability $(1 - p_1 - p_2)$. With 4 degrees of freedom in the form of a_1, a_2, p_1 and p_2, it is possible to normalise the power to

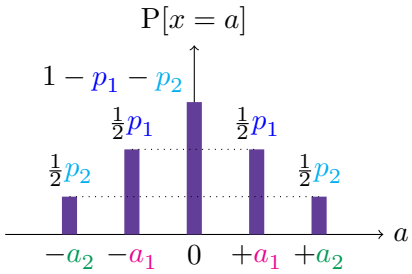


Figure 5.5: Designing the p.m.f. of a discrete symmetric quinary sequence

unity and match $\mathfrak{M}_4$, $\mathfrak{M}_6$ and $\mathfrak{M}_8$ with those of a Gaussian sequence, giving:

$$\mathfrak{M}_2 \triangleq \mathrm{E}\left[\!\left[u^2\right]\!\right] = a_1^2 p_1 + a_2^2 p_2 = 1 \tag{5.6a}$$
$$\mathfrak{M}_4 \triangleq \mathrm{E}\left[\!\left[u^4\right]\!\right] = a_1^4 p_1 + a_2^4 p_2 = 3!! = 3 \tag{5.6b}$$
$$\mathfrak{M}_6 \triangleq \mathrm{E}\left[\!\left[u^6\right]\!\right] = a_1^6 p_1 + a_2^6 p_2 = 5!! = 15 \tag{5.6c}$$
$$\mathfrak{M}_8 \triangleq \mathrm{E}\left[\!\left[u^8\right]\!\right] = a_1^8 p_1 + a_2^8 p_2 = 7!! = 105. \tag{5.6d}$$

Solving the simultaneous equations gives:

$$\begin{aligned} a_1 &= \sqrt{5 - \sqrt{10}}, & a_2 &= \sqrt{5 + \sqrt{10}}, \\ p_1 &= \tfrac{1}{30}\left(7 + 2\sqrt{10}\right) & p_2 &= \tfrac{1}{30}\left(7 - 2\sqrt{10}\right), \end{aligned} \tag{5.7}$$

so that the sequence levels (to four significant figures) are ± 1.356, each with probability 0.2221 and ± 2.857, each with probability 0.01126; and lastly the zero level with a probability of 0.5333. The p.m.f. of this tuned sequence is plotted in Fig. 5.6.

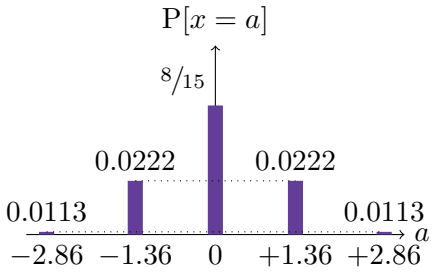


Figure 5.6: The p.m.f. of a symmetric quinary sequence tuned for Gaussianity

5.3 Simulation experiments

Two sets of simulation experiments on systems with a Wiener structure were conducted to confirm the theory developed in Section 5.2. In both experiments, meas-

urements were taken at steady-state.

5.3.1 Experiment 1: Ternary sequences

In this experiment, ternary input sequences are compared with Gaussian input sequences, both having a period $N = 1024$. The ternary sequence experiments were performed for several different values of p_0—the probability that the sequence is at zero (note that $p_0 = (1 - p)$); including $p_0 = 0$ (a binary sequence), $p_0 = 1/3$ (a uniform ternary sequence) and $p_0 = 2/3$ (from (5.3), the value for which $\mathfrak{D}$ should be minimised). p_0 serves as an independent variable to vary the higher order moment of the ternary input in this set of experiments. The outputs were noise-free and were measured for $M = 1024$ independent realisations of each input. The 1024 input and output auto- and cross-power spectra were averaged, with a non-parametric estimate of the BLA then being obtained according to (4.5) with $P = 1$. The averaging is needed even though the system is noise-free because the contribution of the nonlinear distortion terms is different for each realisation as noted in Section 3.1. The linear part of the Wiener system was arbitrarily chosen as a digital first-order low-pass filter with time constant of 4 sampling intervals, i.e. with a transfer function given by $G(z) = z/\left(z - e^{-0.25}\right)$. Two different static nonlinearities were used, as described in the next two subsections.

Although the full analysis could be performed with non-parametric estimates, in order to obtain a more robust estimate of the DF $\mathfrak{D}$, parametric estimates of the BLA were obtained using ELiS, an iterative weighted least squares estimator in the Frequency Domain System Identification Toolbox (*FDIDENT*) (Kollár, 1994). The variances of the complex frequency response estimates were supplied to ELiS for weighting purposes. The cost function used internally was $C = \sum_{k=0}^{N-1} W_k \left|e_k\right|^2$ where $\left|e_k\right|^2 = \left|\hat{Y}[k] - Y[k]\right|^2$ are the squared errors between modelled output spectrum $\hat{Y}$ and actual output spectrum Y in terms of frequency line number k, and W_k are the weighting factors proportional to the reciprocal of the variance at each k. For the Gaussian input sequences, a parametric model of order 1/1 (i.e. one zero, one pole) gave satisfactory fitting, but for the ternary input sequences, a model of order 8/8 was needed because the BLA then contains higher order dynamics than those of the underlying linear system; this is due to nonlinear effects (see example 4.1 on p. 42 of Enqvist, 2005). For the purposes of finding the DF $\mathfrak{D}$, the high order of the parametric model with a ternary input sequence is not of concern.

Cubic nonlinearity

The first nonlinearity was a pure cubic, i.e. $\mathfrak{f}(x) = x^3$, for which it is possible to obtain a theoretical expression for $\mathfrak{D}$ using (3.48). From (5.2b), the deviation of the 4$^{\text{th}}$ moment of the ternary sequence with an arbitrary value of p from that of a Gaussian sequence is given by:

$$\delta_4 = 3!! - a^4 p = 3 - a^4(1 - p_0). \tag{5.8}$$

For the ternary sequence to have a power of unity, $a^2 p = 1$, so that $a^2 = \frac{1}{p} = \frac{1}{1-p_0}$. Substituting in (5.8),

$$\delta_4 = 3 - \frac{1}{1 - p_0}. \tag{5.9}$$

By writing (3.48) and (3.26) in terms of δ_4, the theoretical expression for $\mathfrak{D}$ is given by:

$$\mathfrak{D}_{\text{theory}} = \left(3 - \frac{1}{1 - p_0}\right)^2 \frac{\sum_{k=0}^{\infty} g^6(k)}{9\left[\sum_{k=0}^{\infty} g^2(k)\right]^3}. \tag{5.10}$$

The results are shown in Fig. 5.7. For the cubic nonlinearity, the theoretical Df $\mathfrak{D}_{\text{theory}}$ is shown as a solid line and the simulation results are shown as circles; it can be seen that there is excellent agreement between the two.

Ideal saturation nonlinearity

The second nonlinearity was an ideal saturation characteristic with limits (i.e. clipping levels) set at ± 0.75 and a slope of 1 between these limits. Only simulation results are shown in Fig. 5.7 for this nonlinearity, for which there is currently no theory for finding $\mathfrak{D}$. From these, it can be seen that the minimum value of $\mathfrak{D}$ is obtained with a lower value of p_0 than for the cubic nonlinearity. The difference is not unexpected, because there is no reason to expect that the contributions across multiple higher order terms will be minimised at the same value. It is worth noting that despite this, a lower value of $\mathfrak{D}$ is obtained for this nonlinearity when $p_0 = {}^2/_3$ than for the more widely used values of $p_0 = 0$ (binary input) and ${}^1/_3$ (uniform ternary input).

5.3.2 Experiment 2: Multilevel sequences

In this experiment, Gaussian, binary, ternary, quaternary and quinary input sequences were used; all had a period $N = 2048$. The linearity was a 3-tap FIR filter with transfer function given by $G(z) = (1 + 0.6\,z^{-1} + 0.1\,z^{-2})$, with the relatively short impulse response magnifying the levels of discrepancy (Section 3.5.2)

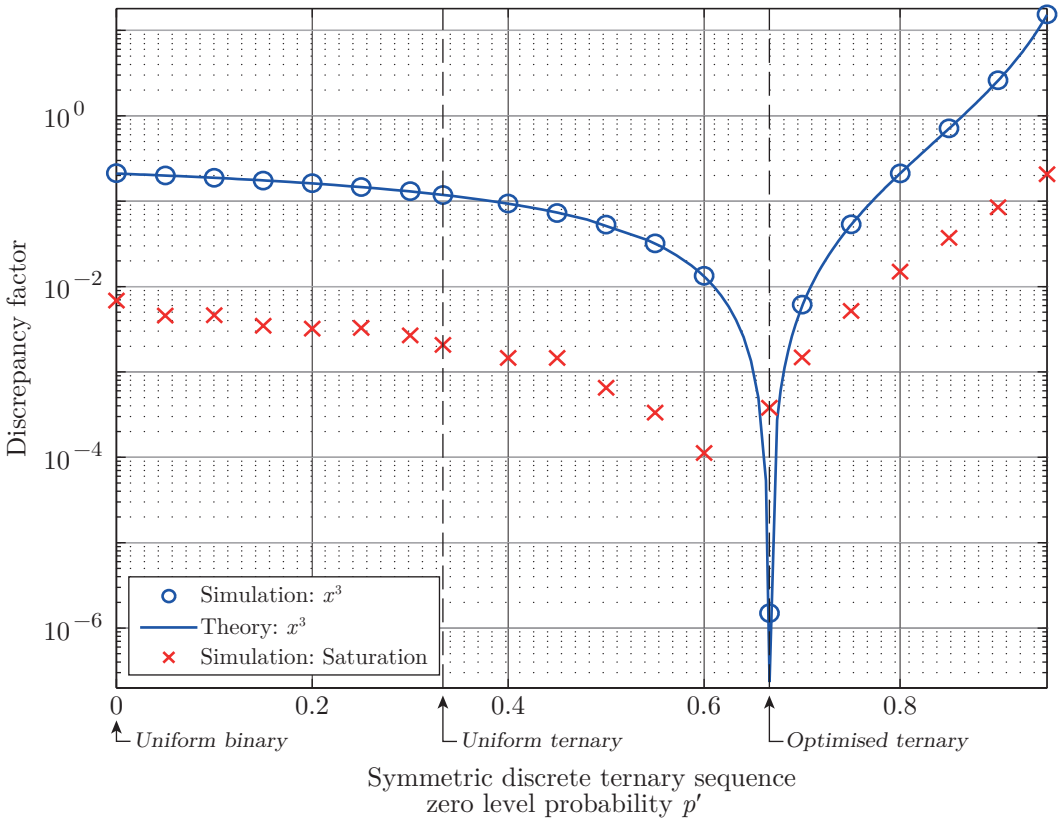


Figure 5.7: Discrepancy Factors of random ternary sequences with various zero-level probabilities

and the actual choice of the filter coefficients here was arbitrary. For the ternary, quaternary and quinary sequences, two versions were created, one with a uniform distribution of the levels and the other with levels and probabilities optimised for Gaussianity—using Equations (5.3) (ternary), (5.5) (quaternary) or (5.7) (quinary). The outputs were noise-free and were measured for 5000 independent realisations of each input. Similar to the method used in Section 5.3.1, the 5000 input and output auto- and cross-spectra were averaged (because the contribution of the non-linear terms is different for each realisation) with a non-parametric estimate of the BLA then being obtained according to (4.5). The nonlinearity was a static polynomial function $\mathfrak{f}(x) = x^7 + x^5 + x^3 + x$; the higher degree than in *Experiment 1* was chosen to emphasise performance differences between sequences with different numbers of levels and between sequences with uniform probability distributions and those optimised for Gaussianity. The MATLAB program code used in this simulation experiment is included in Appendix B.4.

The estimated (frequency domain) BLA magnitude is plotted against fre-

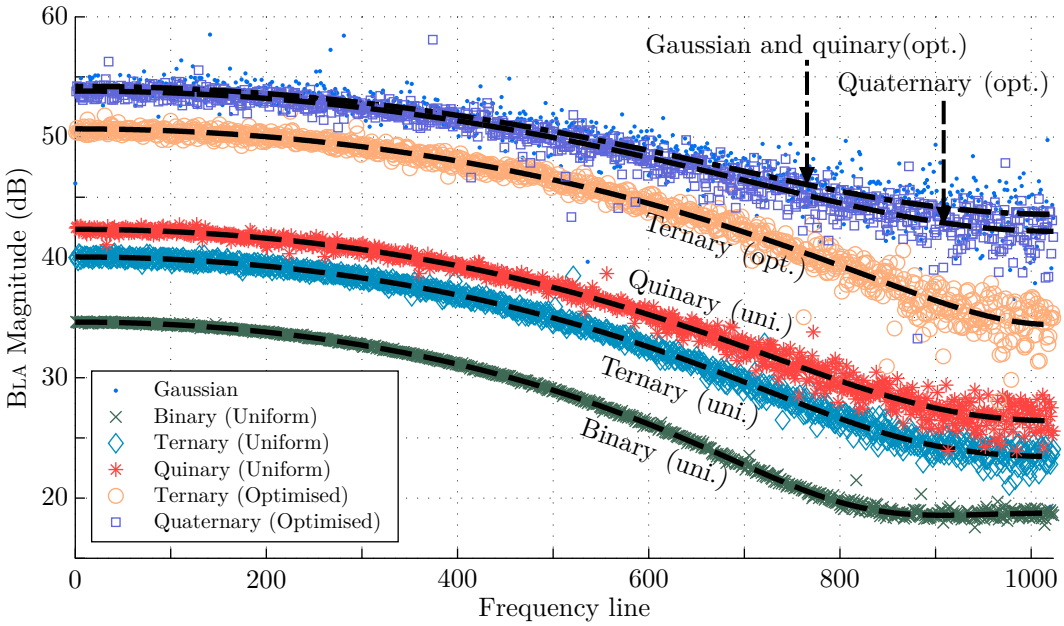


Figure 5.8: Illustration of the difference between BLA's estimated from various multilevel uniform and optimised sequences

quency line for most of these sequences in Fig. 5.8. The top band shows the magnitude estimated using the Gaussian sequences, while the next two show the estimates using the optimised quaternary and ternary sequences. The other three bands show the estimates obtained using quinary, ternary and binary uniformly distributed sequences. The estimates using the quaternary uniformly distributed sequences are not shown, because they come between those for the quinary and ternary uniform bands, which are already relatively close to each other.

The results for the optimised quinary sequences are also not shown in Fig. 5.8, because the BLA estimated from them is indistinguishable from the Gaussian BLA. This is as expected, since from Equations (5.6a) to (5.6d), it is possible to match moments $\mathfrak{M}_m$ up to $m = 8$, which is higher than 7, the highest degree contained in the polynomial nonlinearity. It can be seen that increasing the number of levels of the input sequences decreases the difference between the estimated BLA and that estimated using a Gaussian sequence and that the difference is considerably smaller using an optimised sequence than it is using a uniformly distributed sequence.

5.4 Different identification requirements

With their probability distribution approaching as closely as possible that of a Gaussian sequence, the resulting multilevel sequences are inevitably less suitable for identification of a noisy linear system when there is a limit to the maximum input magnitude (as in many practical situations). For such systems, an input signal with a low crest factor (Cf) (Schoukens, Pintelon, van der Ouderaa & Renneboog, 1988; Pintelon & Schoukens, 2012), correspondingly, a high value of Performance Index for Perturbation Sequences (Pips) (Godfrey, Barker & Tucker, 1999) is desirable. Pips is a compromise between high input sequence power, to maximise the signal-to-noise ratio (Snr) ratio, low input sequence amplitude and to minimise the effects of nonlinearities. Since the moment matching results in the probability distribution of the multilevel sequence becoming more Gaussian in shape, this inevitably reduces the probability of the sequence being at its extreme values, so that the Cf of the multilevel sequence is increased; correspondingly, its Pips is reduced. For example, consider a ternary sequence with levels $\pm a$ and zero then Pips is reduced from $100\sqrt{2/3}\% = 81.7\%$ for a zero-mean sequence with equal probabilities of the three levels to $100\sqrt{1/3}\% = 57.7\%$ for a sequence with the fourth moment matched, as in (5.2b). An acceptable range of values for Pips for identification of a noisy linear system is from 70% to 100% (Barker, Tan & Godfrey, 2009), so for such an application, moment matching would not be used.

This illustrates that an input sequence designed for one purpose may not be the best for a different purpose. This is similar to the choice for multisine signals, where a low value of Cf (high value of Pips) is desirable for the identification of a noisy linear system, whereas a Gaussian multisine (Pintelon & Schoukens, 2012) is desirable for the estimation of the Bla of a nonlinear system.

5.5 Conclusions

This chapter has demonstrated that it is possible to reduce the bias between a Bla estimated from a zero-mean multilevel sequence with levels symmetrical around zero and that estimated from a Gaussian sequence, by matching moments of the multilevel sequence to those of a zero-mean Gaussian sequence. Depending on the dominating degrees of nonlinearity and the number of levels permissible by the input, the bias can be arbitrarily lowered by a suitable design of the probability mass function (p.m.f.) of the random multilevel input sequence. The odd moments of zero-mean sequences are automatically zero, and if the power of the multilevel

sequence is made the same as that of the Gaussian sequence (so matching the second moments), it is possible to match one further moment for a ternary sequence, two for a quaternary sequence and three for a quinary sequence.

However, optimising a multilevel sequence for Gaussianity inevitably increases the crest factor (Cf) and decreases the Performance Index for Perturbation Sequences (Pips) compared to its uniformly distributed counterparts, as mentioned in Section 5.4. As such, in identifying systems with high levels of noise, the identification performance would be compromised somewhat.

Chapter 6

Experiment Verification of the BLA theory

While simulation experiments have confirmed the predictions made for the differences between Best Linear Approximations (BLAs) estimated using Gaussian sequences and those estimated using binary sequences in Chapter 3, experiments based on real physical systems are important to verify the practicality aspect under non-ideal conditions. To this end, a set of experiments was performed on a physical electronic Wiener system to confirm the theory developed for the BLA with respect to the amplitude distribution of the input, specifically, the expressions in Section 3.3.1. The methodology and the results of the physical experiment are described in this chapter.

6.1 Experiment setup

The system was set up using the equipment listed below and the equipment was connected according to the electrical circuit schematic shown in Fig. 6.1. This circuit may be represented by the system diagram of Fig. 6.2 and the structure is equivalent to a Wiener system shown in Fig. 3.2, as the pre-, post- and impedance matching buffers were assumed ideal and introduce no dynamics.

6.1.1 List of equipment

- HP E1401B VXI mainframe with:
  - VXI-MXI-2 interface card

  - Two HP E1430A 10 MSa/s 23-bit analogue-to-digital converters (ADCs), with filtering and memory (hereinafter referred to as the 'acquisition card')
  - HP E1445A arbitrary function generator card
- Desktop computer with PCI-MXI-2 interface card
- Non-inverting pre-buffer with AD8610A op-amp
- Non-inverting post-buffer with TL071CP op-amp
- Two 50 Ω matched impedance measurement buffers
- RC filter circuit with changeable resistors and capacitors
- Pre-built cubic nonlinearity circuit
  - based on AD532JH four-quadrant multipliers
- Tektronix TDS 2001C oscilloscope
- ±12 V and ±15 V power supplies
- A 1.5 nF capacitor with either 2.7 kΩ, 27 kΩ or 110 kΩ resistors in the RC filter,
  - giving cut-off frequency values of $f_{\text{co}} = 39.3\,\text{kHz}, 3.93\,\text{kHz}$ and 0.965 kHz respectively.

The HP VXI mainframe was connected to a desktop computer with the MATLAB software through the VXI-MXI-2 interface card. Data analysis was performed through MATLAB.

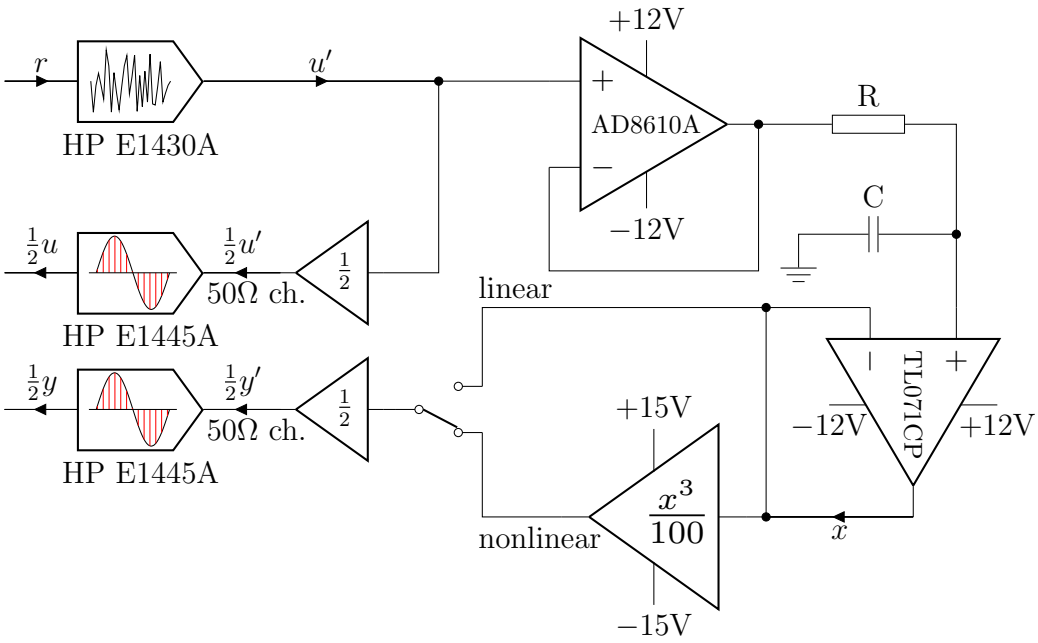


Figure 6.1: Circuit schematic

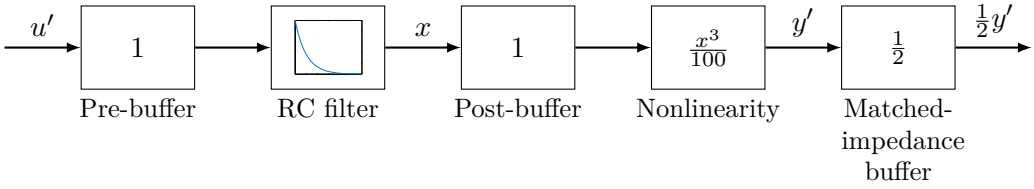


Figure 6.2: Equivalent system structure of Fig. 6.1

6.1.2 Methodology

The objective is to verify theoretical difference, predicted in Section 3.3, between the BLAs obtained from the use of Maximum Length Binary Sequence (MLBS) (Section 2.3.4) and from Gaussian signals, more specifically random-phased multisines (Section 2.3.3). A simple resistor-capacitor (RC) filter element with first order dynamical behaviour was used as the linearity—a RC filter is analogous to a spring-damper system in mechanical terms. For the nonlinearity, a pre-built circuit containing four-quadrant analogue multiplier ICs (integrated circuit elements) that gives the theoretical function $\mathfrak{f}(x) \mapsto 0.01x^3$ was used. Three sets of experiments were performed, each with a different time constant for the linearity. This is implemented by three different resistors; the resistance values used were 2.7 kΩ, 27 kΩ and 110 kΩ. The capacitor value was fixed at 1.5 nF, for a reason which will be explained in Section 6.1.4. The time constant T_p for the first order RC linearity is given by RC, where R is the resistance of the resistor and C is the capacitance of the capacitor.

Table 6.1 lists parameters and their values used in the experimental work. Both types of input were subjected to supersampling (see Section 6.1.4). However, the bandwidth of the multisine was set equal to the clock frequency of the binary sequence. This was performed so that after downsampling (also see Section 6.1.4) the measurement, both types of signal would have identical bandwidth and the spectral whiteness of the two signal types could be preserved—the whiteness of the input spectrum constitutes one of the assumptions of the original discrete time BLA theory in Chapter 3. Since the BLA theory developed so far is valid only for discrete-time systems, no reconstruction filters were used for the waveform generator in order to preserve the zero-order hold (ZOH) nature of u. The focus here is to emulate a discrete-time system as much as possible to verify the BLA theory.

The general procedure of data collection of the experiment for both the linear and nonlinear cases was as follows:

1. Generate the reference signal r, either:
   (a) a discrete random-phased multisine for the Gaussian case, or:
   (b) an MLBS of period $N_\mathrm{base} = 511$ samples for the binary case.

Table 6.1: Table of parameters and settings for the verification experiment

Symbol	Description	Value	(unit)
f_s	Sampling frequency for the arbitrary waveform generator, and acquisition cards. The Nyquist frequency is then $f_s/2$.	312.5	kHz
T_s	Sampling interval $= 1/f_s$.	3.2	µs
μ	Over-sampling ratio for MLBS's (see Section 6.1.4).	8	
f_c	Clock frequency of the MLBS's $(= f_s/\mu)$.	$39\,1/16$	kHz
T_b	Bit interval for the MLBS's $(= 1/f_c)$.	25.6	µs
f_w	Bandwidth of the multisine sequence after downsampling $(= f_c = f_s/\mu)$.	$39\,1/16$	kHz
f_{aa}	Anti-aliasing filter cut-off frequency $\equiv 0.4 f_s$. This coupling with the sampling frequency value is internally enforced by the HP1430A acquisition cards.	125	kHz
N_{base}	Base length of sequence after subsampling, (= length of a 9-tap m-sequence)	511	Sa†
N	Length of a data record $(= \mu N_{base})$.	4088	Sa†
P	Number of steady-state periods measured (linear case; nonlinear case).	12; 4	
M	Number of independent realisations (linear case; nonlinear case).	5; 16	
V_{rms}	Root mean squared (rms) voltage of the input signals.	1.5	V

†Sa = samples

2. Realise the periodic signal using the HP E1445A arbitrary function generator card. The excitation is uninterrupted and continuously turned on from this point onwards.
3. Pause for 5 seconds so that transient effects in the measurements are expected to be negligible.
4. Initiate measurements with the acquisition cards and collect P periods of data. The measurement intervals are internally synchronised with the generator.
5. Due to internal attenuation of the matched impedance buffers, measurements are normalised by a factor of two to obtain u and y (in multiple periods).
6. Go to Step 1 and repeat for a different realisation of input until M data sets of different input realisations are obtained.

Note that across the M sub-experiments, the same input sequence realisation was never used more than once.

6.1.3 Robust non-parametric identification procedure

From Section 4.2, given a set of input and output data from M sub-experiments of independent input realisations, each with P periods of steady-state measurements, a robust estimator for the BLA is given by:

$$\hat{G}_{\text{BLA}}(\jmath\omega) \triangleq \frac{\sum_{m=1}^{M}\sum_{p=1}^{P} Y^{[m,p]}(\jmath\omega)\cdot\overline{U}^{[m,p]}(\jmath\omega)}{P\sum_{m=1}^{M} U^{[m,p]}(\jmath\omega)\cdot\overline{U}^{[m,p]}(\jmath\omega)} \tag{6.1}$$

where $U^{[m,p]}$ and $Y^{[m,p]}$ contain the m^{th} sub-experiment and p^{th} period of the measured input spectrum $U(\jmath\omega)$ and the measured output spectrum $Y(\jmath\omega)$. The bar on top of the spectra variables denotes the complex conjugate. For reasons stated in Section 6.1.4, the measured input spectrum for the binary excitation case, $U^{[m,\forall p]}$ was taken as the reference input spectrum $R^{[m]}(\jmath\omega)$.

The estimator is robust against noise disturbances and is unbiased if input noise levels are small.

6.1.4 Supersampling

The HP VXI system is capable of any sampling frequency up to and including 10 MHz. In the experiment, the measurements were oversampled by a factor of μ above the clock frequency of the binary sequence input f_{c}. The upper frequency of the bandwidth of the random-phased multisine f_{w} was set equal to f_{c}. The measurement data were aligned and then subjected to downsampling by the same factor μ, procedures for which are described below. After downsampling, both input signal types had identical bandwidth. The supersampling and the subsequent downsampling were performed for two main reasons detailed in the next two subsections (titled 'Ringing and overshoots' and 'Anti-alias').

The RC filter has low-pass (smoothing) characteristics. To downsample (or subsample) the measurement, a location of the highest peak (or lowest trough) of the output signal was taken as the reference point. From this reference point onwards and backwards every μ^{th} sample was taken as an idealised ZOH measurement, with the ZOH clock frequency a factor μ lower than the original sampling (hence a downsampling). The reference signal r and the measured output y were also aligned through this reference point, so that the peaks of the output after RC filtering would then occur directly after the switching points of the binary reference input. The procedure is more easily appreciated by referring to Fig. 6.3 by comparing the reference

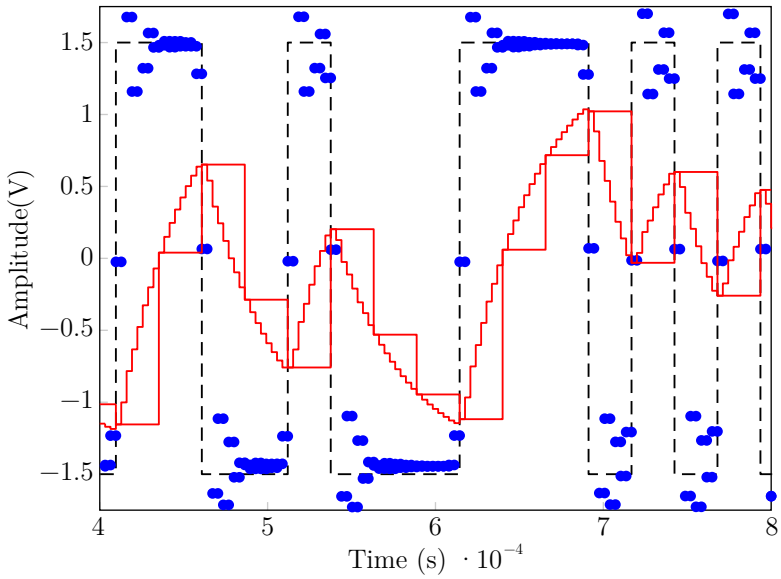


Figure 6.3: The use of supersampling and subsampling – Reference input (dashed line), Measured input (dots), Measured output: a) supersampled and b) subsampled (solid lines).

input (dashed line) and the two solid lines representing the original high frequency sampling and the subsequent downsampled and aligned ZOH data. If there were no ringing, overshoot, nonlinear effects or noise, this subsampling procedure would result in a perfect reconstruction of the behaviour of an ideal ZOH sampler according to discrete-time theory and this has been verified by simulation. This subsampling procedure was performed for both linear and nonlinear measurements. Since this procedure can only be reliably performed through the easily visible binary switching points, the same alignment amount and the time coordinate of the reference point were used for the corresponding case with multisine inputs.

Ringing and overshoots

During testing with binary excitation signals, it was observed through the oscilloscope that all operational amplifier (op-amp) based electronic buffers introduce high frequency oscillations in form of ringing to a varying extent. This was caused by non-ideal step-response characteristics when load or parasitic capacitances at output of op-amps introduce unintended poles in the transfer characteristics of the op-amps through feedback. The datasheets of many op-amps have step-response graphs that illustrate this.

In this experiment setup, the overshoots and undershoots were especially large, up to 20% of the step size with the pre-buffer due to the capacitive load at the RC circuit, even when a higher quality op-amp (with regards to its ability in driving capacitive loads) was used (Analog Devices, 2008). The overshoot depends on the load or parasitic capacitance hence the load capacitor C of the RC circuit was fixed at 1.5 nF for consistency.

Moreover, the HP E1430A acquisition cards themselves have significant overshoots that can be seen in the measurement data, although the oscilloscope suggested the actual acquisition inputs u' and y' were relatively free of such effects. This may be caused by the high order high cut-off frequency anti-aliasing filter having oscillatory step responses. The ringing at the measured input channel from an acquisition card can be seen in Fig. 6.3. This phenomenon persisted with an Agilent 33120A waveform generator directly driving the acquisition cards, isolated completely from the system in question.

While the RC passively formed a low pass filter and was capable of minimising the effect of ringing from the pre-buffer, overshoot and ringing from the acquisition cards were inevitable. Due to the nature of sample-and-hold at the acquisition cards, the use of supersampling was necessary to obtain measurements of acceptable accuracy. The BLA theory developed is incapable of modelling in continuous time domain of such effects at the moment. For multisine input sequences, there are no noticeable ringing or overshoot effects.

Because of the overshoot and ringing present in the measurement data from the HP E1430A acquisition cards, the signal sequence u is no longer reliable and accurate representation of u' in the MLBS case. Henceforth in dealing with binary sequences, the reference signal r is used as the basis for identification.

In addition, manual alignment of the measured input and output signals can be performed.

Anti-alias

It is necessary to minimise the effect of anti-aliasing filters on the measurements because of the use of the ideal reference signal r instead of measured input u in the case of binary excitations. In addition, the nonlinearity broadens the bandwidth of the output, which then may be interfered with by the anti-aliasing filter if action is not taken. Supersampling allows the internal anti-aliasing filter to be bypassed since the internal anti-aliasing filter of the HP E1430A acquisition cards have their cut-off frequencies f_{aa} dependent upon the sampling frequency f_{s} (see Table 6.1). The combination of the specified low bandwidth of the multisine, the discrete nature

of binary excitation signals and the low pass characteristics of RC mean that any real aliasing effect was negligible. It has been shown that broadening of spectrum due to nonlinearity would result in aliased components that are never coherent with the original input component (Pintelon & Schoukens, 2012, theorem 3.21), hence the lack of anti-aliasing filter would only act as additional uncorrelated noise in the BLA measurement.

6.1.5 Linear measurements

Measurements were performed to identify either a parametric or a non-parametric model for the linearity. The rms signal amplitudes for the Gaussian and binary signals were both set to 1.5 V. The non-parametric model was obtained using (6.1), and a parametric model was fitted where suitable using the iterative weighted non-linear least squares procedure provided by the Estimator for Linear Systems (ELiS) tool in the *FDIDENT* Toolbox for MATLAB (Kollár, 1994). The weighting factors were proportional to the reciprocal of the variances at each frequency point. The isolation provided by the pre-buffer and post-buffer for the RC circuit introduced some additional linear dynamics, and hence suitable single pole models could not be fitted to the data. When a parametric model of order greater than four was not sufficient to describe the transfer characteristics of the linearity in both the z-domain and the s-domain, the non-parametric model was used. This was the case for when the resistor value was 110 kΩ, and hence Fig. 6.7 does not contain the results from the parametric model.

Fig. 6.4 shows the result of a non-parametric identification of the linearity. The noise variances, estimated using (4.6), indicate levels of exogenous additive noise from the environment whereas the total variances, estimated using (4.9), indicate the levels of nonlinear distortions plus environment noise. There is a discrepancy between the result obtained with multisine sequences and that obtained from MLBS's. This suggests input dependent nonlinear characteristics which include slew rate limitations of the op-amps and nonlinear effects from ringing oscillations.

As an example, for $R = 27\,\text{k}\Omega$ and $C = 1.5\,\text{nF}$, the time constant $T_\text{p} = 27{\times}10^3 \cdot 1.5{\times}10^{-9} = 4.05{\times}10^{-5}$ seconds. With a sampling interval T_c given by $1/f_\text{c} = (39^1/_{16} \times 10^3)^{-1}$ seconds, $T_\text{c}/T_\text{p} = 0.6321$, and therefore the theoretical transfer function is:

$$G(z) = \frac{z}{z - e^{-0.6321}} = \frac{z}{z - 0.5315}. \tag{6.2}$$

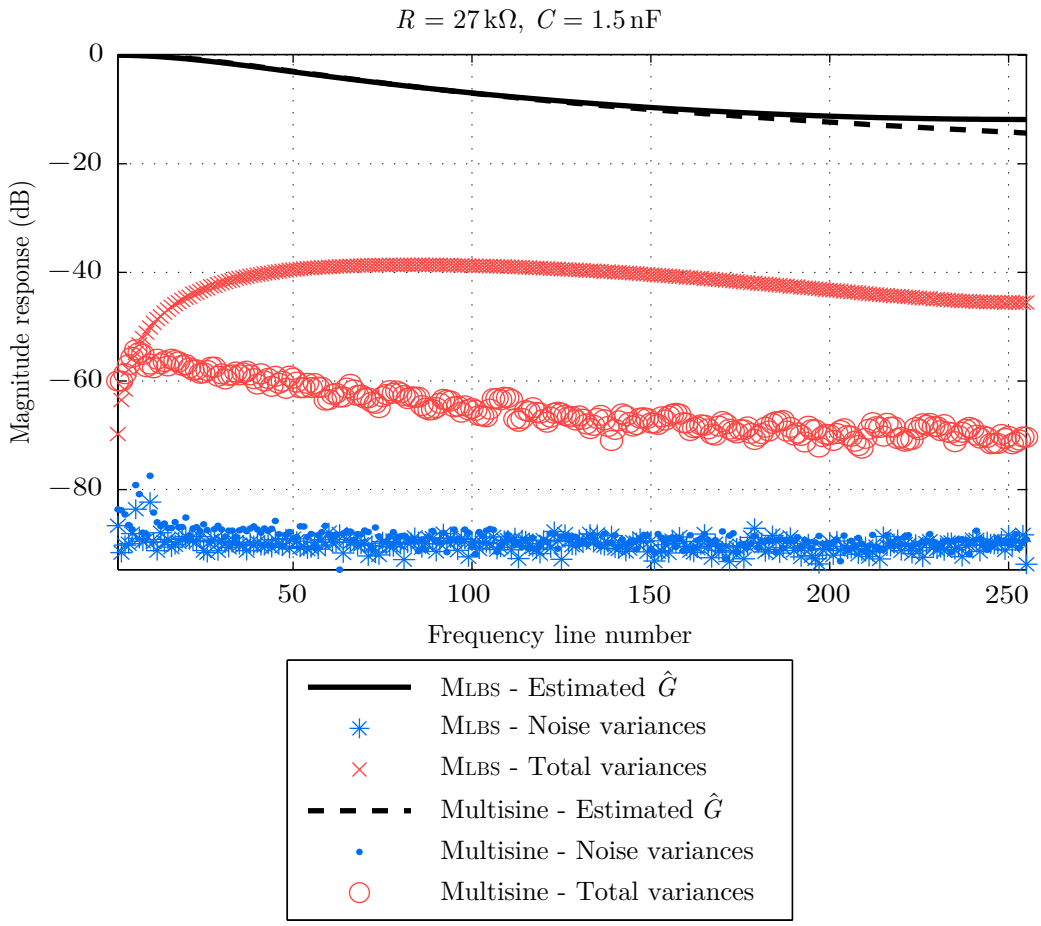


Figure 6.4: RC Linearity identification with op-amp based pre- and post-buffers. (To convert to frequency (Hz), line number should be multiplied by $f_s/(8 \times 511)$)

The parametric model identified for the MLBS case with sampling time T_c was:

$$\hat{G}(z) = \frac{0.01011(z+41.23)(z+0.05967)}{(z-0.5481)(z+0.01818)}. \tag{6.3}$$

It can be seen that the estimated positive pole is very close to the theoretical value, but as noted above, the pre-buffer and the post-buffer to the RC circuit introduced some additional dynamics, with a negative zero and a negative pole very close to the origin, and a further negative zero that is so large that it can be regarded as a constant over the frequency range of interest.

Despite the fact that the system-under-test was linear, there were nonlinear distortions in both input cases and the level was higher for the binary input. This was due to nonlinear effects from the unity-gain op-amp buffers, especially from

the pre-buffer, which had to drive the capacitive load. Ringing oscillations were especially noticeable with binary inputs (see Section 6.1.4). If the buffers were not used, the nonlinear distortions disappear regardless of the input signal. However, due to current driving limitation of the signal generator and the capacitive load, there was unacceptable distortion of the realised input for the binary case, hence the buffers were necessary.

6.1.6 Nonlinear measurements and the BLA theory

The nonlinear measurements were obtained in a similar manner to the linear measurement case. The non-parametric BLA was obtained using (6.1).

To enable comparison with the theory, additional information is required. This includes the higher order even moments of the input signals, the signal power (or the rms value V_{rms}), the impulse response of the linearity and the polynomial coefficients of the nonlinearity. For the Gaussian case, the even order moments of u (i.e., $\mathrm{E}[\![u^n]\!]$) were measured and averaged for a single experiment, for even n. For the binary case, u was replaced by r hence $\mathrm{E}[\![r^n]\!] = V_{\text{rms}}^n$. The impulse response function (IRF) of the linearity was taken from the parametric model if available, and the non-parametric model by inverse Fourier transform (FT). Finally, the nonlinearity was isolated from the rest of the system and independently identified—a periodic random-phased multisine excitation with an rms power of 1.5 V was directly introduced at the input (x) of the nonlinearity and the input and output from the nonlinearity were measured for 20 periods. A least squares polynomial regression based on the "`polyfit`" routine provided by MATLAB was applied to the 20 periods of the output and input data. The polynomial fitted to the nonlinearity was:

$$\mathfrak{f}_{\text{NL}}(x) = 0.01088x^3 - 0.001356x^2 + 0.008169x + 0.05816. \tag{6.4}$$

6.2 Results and analysis

Figures 6.5, 6.6 and 6.7 show the comparison of the BLA obtained through experiment results and those obtained from theory, for a Wiener system with (non-ideal, see (6.4)) cubic nonlinearity and RC filter linearity with $C = 1.5\,\text{nF}$ for all three cases and $R = 2.7\,\text{k}\Omega$, $27\,\text{k}\Omega$ and $110\,\text{k}\Omega$ respectively.

With $R = 2.7\,\text{k}\Omega$, the RC filter has a cut-off (or corner) frequency of $f_{\text{co}} = \frac{1}{2\pi RC} \approx 39.3\,\text{kHz}$ and acts as an all-pass filter since the binary signal clock frequency was $f_{\text{c}} = 39^{1}/_{16}\,\text{kHz}$. Unfortunately this means ringing and overshoot effects (see Section 6.1.4) were significant immediately after the RC filter stage. The identifica-

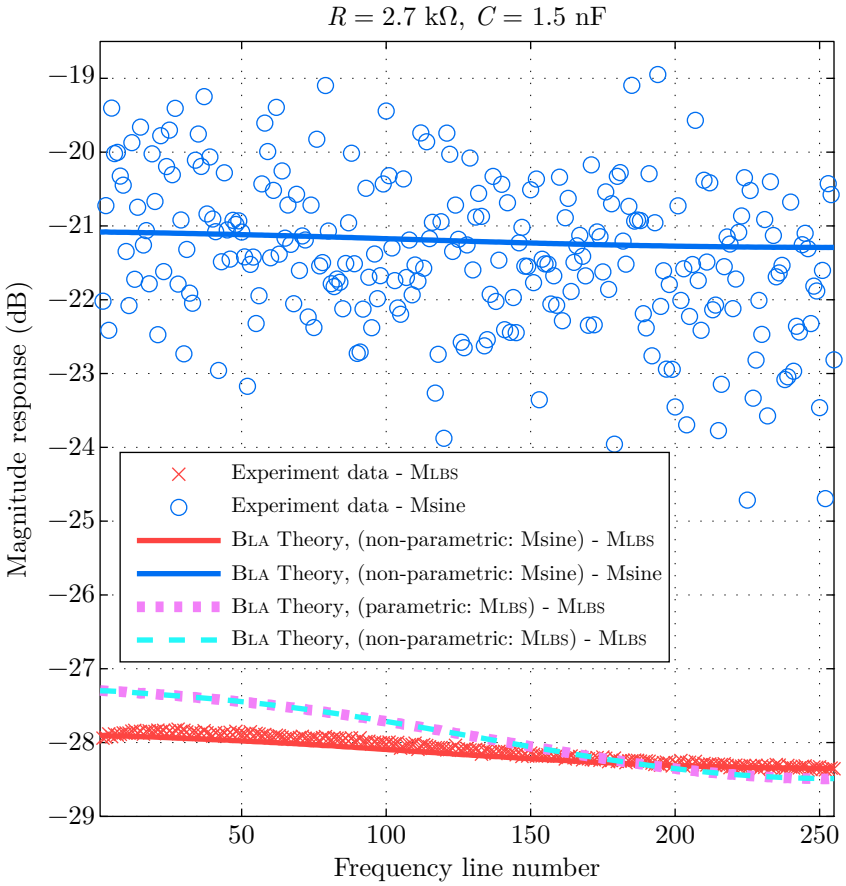


Figure 6.5: Experiment result of the identification of the BLA with Gaussian and binary inputs for an electronic Wiener system with non-ideal cubic nonlinearity and a RC linearity with $R = 2.7\text{k}\Omega$, $C = 1.5\,\text{nF}$ giving corner frequency $f_{\text{co}} = 39.3\text{kHz}$. (For conversion from frequency line number to Hz, see caption of Fig. 6.4.)

tion of the linearity using MLBS's would yield unreliable results despite subsampling techniques. Here the use of non-parametric models of the linearity identified by a multisine was more suitable for the BLA theory. This can be seen by the fact that in Fig. 6.5 the solid red line, representing the BLA theory based on a non-parametric model of the linearity identified with a multisine sequence, was able to match the experiment data represented by crosses more closely than that based on a model of the linearity (parametric or non-parametric) identified with an MLBS. There are minimal differences between results derived from the non-parametric and parametric models of the linearity—the plots (cyan dashed and magenta heavy-dotted, respectively) are very close to each other.

For $R = 27\,\text{k}\Omega$, the RC filter had a corner frequency of approximately

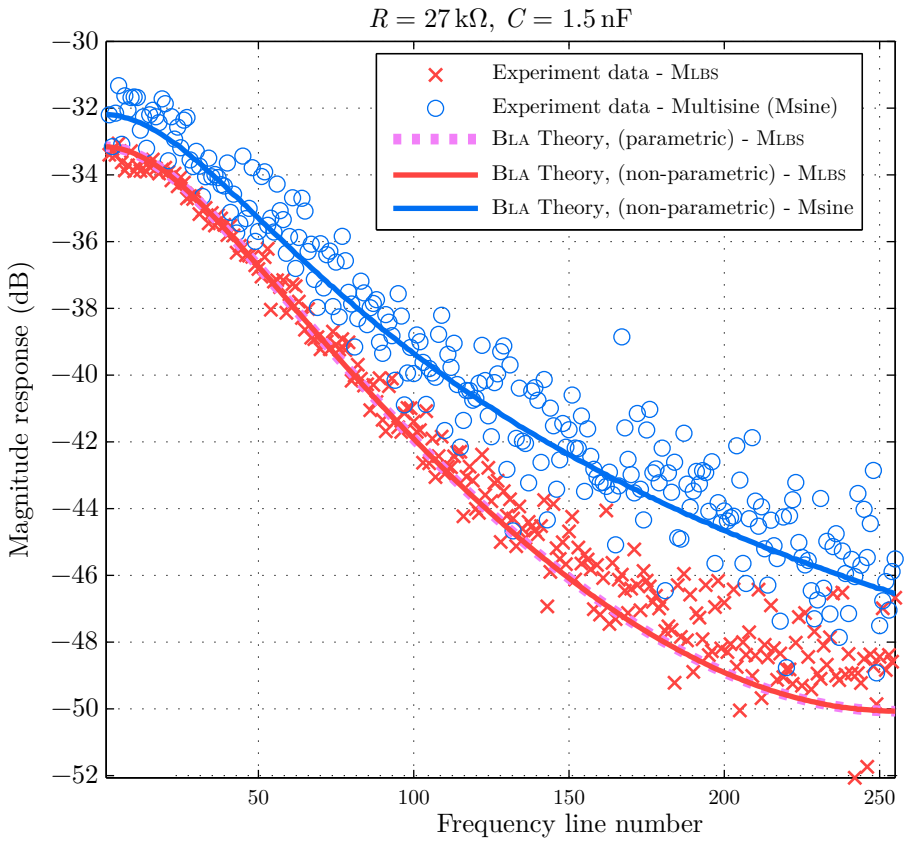


Figure 6.6: As Fig. 6.5, but with $R = 27\,\mathrm{k\Omega}$, giving corner frequency $f_{\mathrm{co}} = 3.93\,\mathrm{kHz}$.

3.93 kHz. Ringing and overshoot effects were then negligible immediately after the RC filter stage. Here the non-parametric models of the linearity for the MLBS and multisine were used for their respective counterparts. In addition, the parametric model of (6.3) from Section 6.1.5 was used in the BLA theory to calculate the biased theoretical BLA for binary sequences. There were no discernible differences in the BLA theory calculated from the non-parametric and parametric models of the linearity as shown by the overlapping of the heavy-dotted magenta line and the solid red line in Fig. 6.6.

When $R = 110\,\mathrm{k\Omega}$, the RC filter had a corner frequency of about 0.965 kHz. The result is illustrated in Fig. 6.7. This time parametric models up to order four produced by ELiS could not produce a fit of adequate quality to the transfer function of the linearity. Nevertheless the BLA theory based on the non-parametric model was able to match the experiment data in both the gain and the shape of the transfer characteristics.

As the time constant of the system increases, the length of the impulse re-

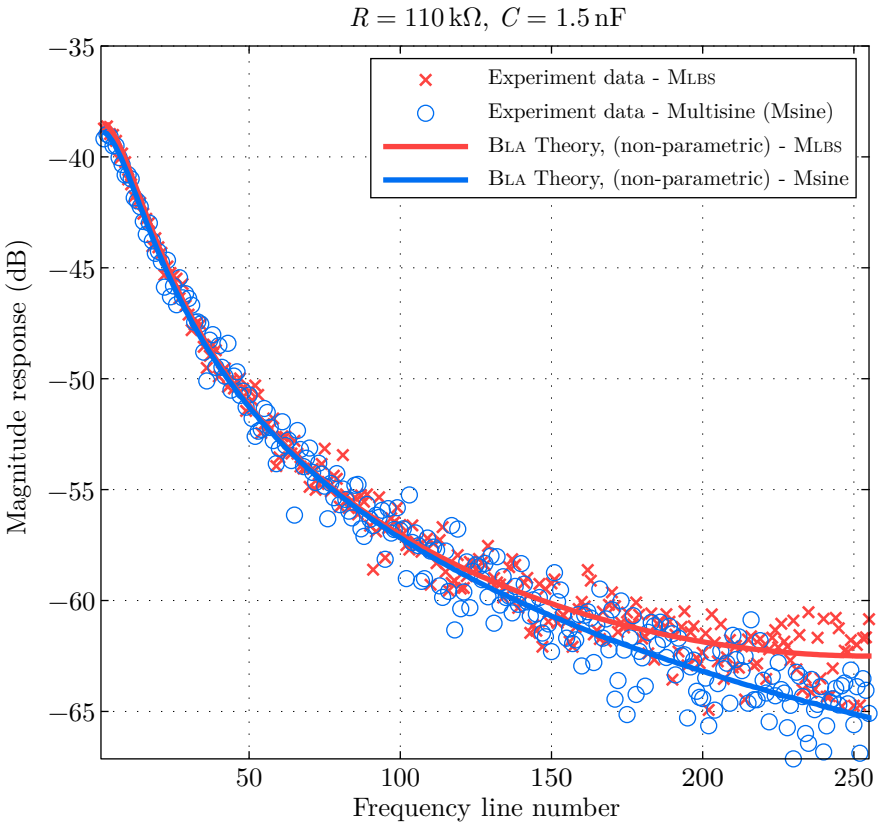


Figure 6.7: As Fig. 6.5, but with $R = 110\,\mathrm{k}\Omega$, giving corner frequency $f_{\mathrm{co}} = 0.965\,\mathrm{kHz}$.

sponse of the system increases. It has been shown in Section 3.5.2 that this results in a BLA estimated by a signal with an arbitrary amplitude distribution converging to that obtained from Gaussian signal. This is also observed in Figures 6.5 to 6.7.

6.3 Conclusions

For all three sets of experiments investigated, it can be seen that predictions from the BLA theory and experiment results are in good agreement. This is despite the fact that the BLA theory was based on IRFs of the linearity modelled as finite impulse response (FIR) filters, whereas here the RC filter circuit is an infinite impulse response (IIR) filter. This chapter successfully verifies the theoretical work of Chapter 3 and part of Chapter 4.

The difficulties encountered with the experiment, mainly the ringing and overshoot effects, illustrate a weakness in the z-domain discrete-time theory. It may therefore be beneficial to extend the theory to the continuous-time s-domain. While

beyond the scope of this thesis, work performed on finding closed-form solutions of the BLA theory in the continuous-time domain suggests far more complex expressions instead of the relatively simple results seen for the discrete-time counterparts in Chapter 3, for even 3^{rd} degree nonlinearities.

Chapter 7

Benchmark Study: Identification of a Wiener-Hammerstein System using an Incremental Nonlinear Optimisation Technique

This chapter documents a two-stage iterative nonlinear optimisation algorithm identifying both the linearity and the static nonlinearity of a Wiener-Hammerstein (WH) system (Tan, Wong & Godfrey, 2012), in a benchmark problem that was first proposed in a special session of the 15th International Federation of Automatic Control (IFAC) Symposium on System Identification (SYSID) 2009 (Schoukens, Suykens & Ljung, 2009) and it is referred to as the *Benchmark* throughout this chapter. *Disclaimer*: The work in this chapter was a collaboration with Dr. Tan, Ai Hui of the Multimedia University, Malaysia, performed in 2010 and 2011. She was also the first author of the related and published paper (Tan, Wong & Godfrey, 2012). The author of this thesis was responsible for the data analysis and applying the optimisation routine for all the combinations of zeros and poles tried.

The method in this chapter decomposes the Gaussian Best Linear Approximation (BLA), which contains the combined linear dynamics of the two linearity blocks (Pintelon & Schoukens, 2012; Schoukens, Lataire et al., 2009), through brute-force factorisation maintaining realisable filters and hence causality and using the decomposed transfer functions as initial estimates for the linearities. All possible combinations of these factorisations were tested by trial and error though a two-stage iterative algorithm which updates the estimates of both the linearities and the

static nonlinearity in turn. Through many iterations the best model was chosen. The static nonlinearity was modelled firstly by simple polynomial and then dual-polynomial functions, with the latter reducing the number of parameters required and the accuracy of the model, due to the nature of a break-point in the saturation characteristics of the underlying true nonlinearity.

The proposed approach not only has a small number of parameters compared with many state-space models, and in spite of the small parameter space, it has achieved excellent performance in terms of mean squared error (MSE) in validation.

7.1 Problem definition

It is assumed the structure of the system is known to be a WH system with a static nonlinearity $\mathfrak{f}(x)$ sandwiched by two linearities, as Fig. 7.1 depicts. Apart from the linearities being realisable and the nonlinearity being static, the underlying physics of the system are assumed unknown, i.e. this is a blackbox modelling approach with model structure imposed.

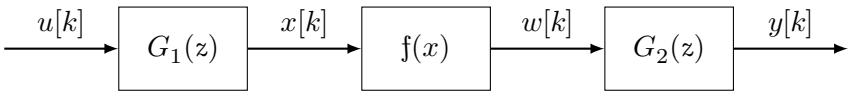


Figure 7.1: Wiener-Hammerstein system structure

7.1.1 The real system

Benchmark data were obtained from a physical electronic system. The first (preceding) linear block $G_1(z)$ was a third order Chebyshev Type-I (equiripple in pass-band) low-pass filter with a cut-off frequency of 4.4 kHz; the nonlinearity was a diode circuit emulating saturation characteristics; finally, the second (succeeding) linear block $G_2(z)$ was a third order inverse (Type-II) Chebyshev filter with a stop-band attenuation of 40 dB, starting at 5 kHz. The input u was obtained from a zero-order hold (ZOH) arbitrary waveform generator, followed by an analogue low-pass reconstruction filter with a cut-off frequency of 10 kHz. Records of the input u and output y provided consisted of 188 000 samples with a sampling frequency of 51.2 kHz, giving an overall record length of 3.671 875 s.

7.2 Benchmark metric

It is stated in the *Benchmark* specification that the first 100 000 samples, referred to as the training set throughout this chapter, are available for the model estimation. The model error on training set however, should only be calculated from sample 1001 to 100 000. The validation set contains samples from 100 001 to 188 000, but the first 1000 samples in this set are again omitted for error calculation to minimise influence from transients.

If the model errors are denoted $e[k]$, the root mean squared (rms) training error is given by:

$$e_{\mathrm{rms,T}}[k] \triangleq \sqrt{\frac{1}{99000} \sum_{k=1001}^{100\,000} e^2[k]} \tag{7.1}$$

and the validation error by:

$$e_{\mathrm{rms,V}}[k] \triangleq \sqrt{\frac{1}{87000} \sum_{k=101\,001}^{188\,000} e^2[k]} \tag{7.2}$$

This rms validation error is used as the metric for comparisons with other papers. In addition, for the convenience of discussion, the mean μ and the standard deviation σ of the simulation errors $e[k]$ are defined, for the training set as:

$$\begin{cases} \mu_{\mathrm{T}} \triangleq \dfrac{1}{99\,000} \displaystyle\sum_{k=1001}^{100\,000} e[k] & \text{(7.3a)} \\ \sigma_{\mathrm{T}} \triangleq \sqrt{\dfrac{1}{99\,000} \displaystyle\sum_{k=1001}^{100\,000} \big(e[k] - \mu_{\mathrm{T}}\big)^2} & \text{(7.3b)} \end{cases}$$

and for the validation set as:

$$\begin{cases} \mu_{\mathrm{V}} \triangleq \dfrac{1}{87\,000} \displaystyle\sum_{k=101001}^{188\,000} e[k] & \text{(7.4a)} \\ \sigma_{\mathrm{V}} \triangleq \sqrt{\dfrac{1}{87\,000} \displaystyle\sum_{k=101\,001}^{188\,000} \big(e[k] - \mu_{\mathrm{V}}\big)^2} & \text{(7.4b)} \end{cases}$$

7.3 Identification of the BLA

The BLA of the overall system, $G_{\mathrm{BLA}}(z^{-1})$, was estimated using maximum likelihood estimation (Schoukens, Pintelon & Renneboog, 1988) in the frequency domain and implemented using the ELiS tool provided by the MATLAB Frequency Domain

System Identification Toolbox (*FDIDENT*) (Kollár, 1994). The input to the system was a low-pass filtered Gaussian signal with cut-off frequency of 10 kHz and a sampling frequency of 51.2 kHz. With 100 000 data points, the frequency resolution is 0.512 Hz. The highest harmonic used for estimation was set to 19 500, corresponding to a frequency of 9.984 kHz, which is slightly below the cut-off frequency. The variance of the input measurement was not provided in the benchmark dataset, and hence the value supplied to ELiS was set to zero. The output noise variance was set to a constant. The Newton-Gauss optimisation algorithm was used.

Model orders from 1 to 7 for both the number of poles and zeros were tried. The final model order was selected based on the model giving the minimal training set rms error $e_{\mathrm{rms,T}}$, as specified in (7.1), and the Akaike Information criterion (AIC) (Hirotugu, 1974), calculated by ELiS to indicate model over-fitting.

The resulting model order with minimal $e_{\mathrm{rms,T}}$ and a good AIC was found to have 6 zeros and 6 poles, with the BLA given by $G_{\mathrm{BLA,A}}$:

$$G_{\mathrm{BLA,A}}(z^{-1}) = -1.4362\times10^{-3} \cdot \frac{(1-2.4132\,z^{-1})(1-0.9356\,z^{-1})}{(1-0.9325\,z^{-1})(1-0.8649\,z^{-1})} \cdot \frac{(1-1.5518\,z^{-1}+1.0411\,z^{-2})(1-1.1501\,z^{-1}+1.2708\,z^{-2})}{(1-1.7902\,z^{-1}+0.8296\,z^{-2})(1-1.3919\,z^{-1}+0.6483\,z^{-2})}. \quad (7.5)$$

This particular BLA gives a training set simulation error of $e_{\mathrm{rms,T}} = 5.53\times10^{-2}$, while that for the validation set is $e_{\mathrm{rms,V}} = 5.62\times10^{-2}$. Note that generating a discrete-time model for a continuous-time system only gives an approximation and is valid only for this particular sampling frequency of 51.2 kHz.

With $G_{\mathrm{BLA,A}}$, using (7.3), $\mu_{\mathrm{T}} = 3.62\times10^{-2}$ and $\sigma_{\mathrm{T}} = 4.22\times10^{-2}$ and using (7.4), $\mu_{\mathrm{V}} = 3.59\times10^{-2}$ and $\sigma_{\mathrm{V}} = 4.32\times10^{-2}$.

$G_{\mathrm{BLA,A}}$ is designated with subscript 'A' because this estimation is reconsidered due to concerns on the accuracy of the high frequency performance. The higher frequency range is likely to be distorted by the nonlinearity due to its low gain. As noted in Section 7.1.1, the second linear block is known to have a stop-band with attenuation greater than 40 dB, starting at 5 kHz. If a linear model is to be applied as a sub-block(s) along with another nonlinear block, this higher frequency range should be omitted in the identification. To this end, the frequency range selected for the identification step was reduced to cover harmonics from 1 to 8000 only, with harmonic 8000 corresponding to a frequency of 4.096 kHz. At this frequency, the output power has decreased to about −30 dB from the peak gain. The resulting

BLA is:

$$G_{\mathrm{BLA,B}}(z^{-1}) = -2.2459\times10^{-2} \cdot \frac{(1-1.604\,z^{-1})(1-0.9727\,z^{-1})}{(1-0.9724\,z^{-1})(1-0.8521\,z^{-1})} \cdot \frac{(1-1.8757\,z^{-1}+0.9417\,z^{-2})(1-2.0212\,z^{-1}+1.3604\,z^{-2})}{(1-1.7896\,z^{-1}+0.8291\,z^{-2})(1-1.8761\,z^{-1}+0.9423\,z^{-2})} \quad (7.6)$$

and the performance measures were $e_{\mathrm{rms,T}} = 5.68\times10^{-2}$, $e_{\mathrm{rms,V}} = 5.77\times10^{-2}$, $\mu_{\mathrm{T}} = 3.62\times10^{-2}$ and $\sigma_{\mathrm{T}} = 4.41\times10^{-2}$, $\mu_{\mathrm{V}} = 3.59\times10^{-2}$ and $\sigma_{\mathrm{V}} = 4.51\times10^{-2}$. Although all these except μ_{T} and μ_{V} were all higher than those for $G_{\mathrm{BLA,A}}$, it is expected that this will yield better results when the nonlinear block is added.

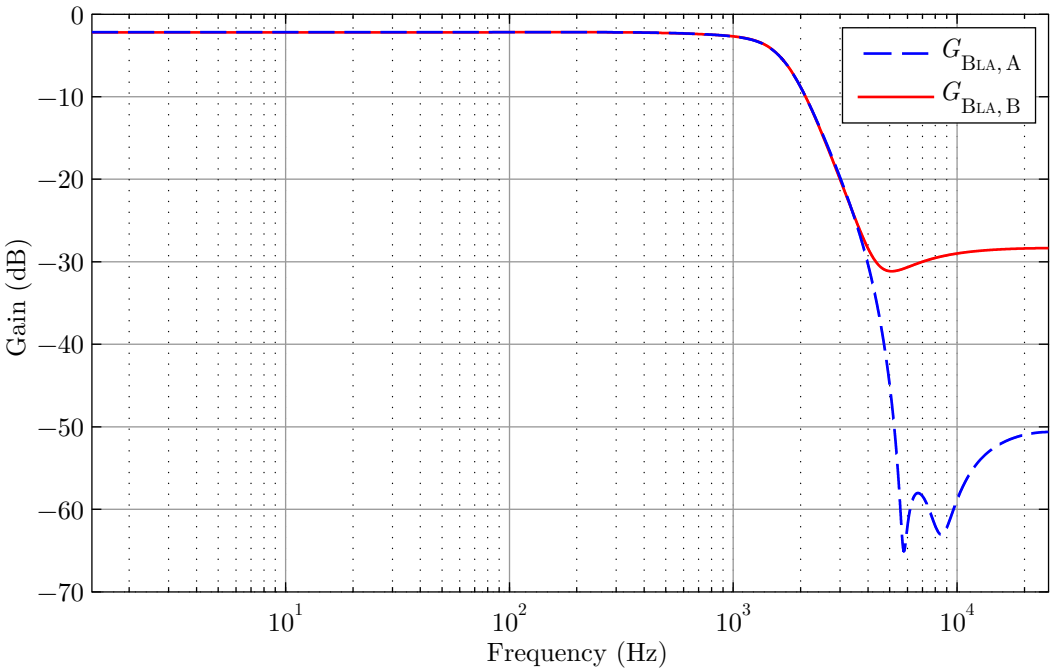


Figure 7.2: Bode plot comparing the high frequency fitting of $G_{\mathrm{BLA,A}}$ and $G_{\mathrm{BLA,B}}$

Fig. 7.2 is a Bode plot showing the magnitude frequency responses of both $G_{\mathrm{BLA,A}}$ and $G_{\mathrm{BLA,B}}$ plotted against each other. In the figure one can see while the high frequency components are not fitted by $G_{\mathrm{BLA,B}}$, the low frequency components are well matched.

7.3.1 Incremental nonlinear optimisation technique

The BLA identified in the previous section models the output of the WH system structure given in Fig. 7.1 by replacing it with a purely linear system with the transfer characteristics described by the BLA. While this linearisation may produce

an acceptable model in some applications, it is nevertheless a linear model incapable of modelling nonlinear effects such as the disproportionate gain when input power is increased for example. To reduce the errors further, it is necessary to consider a nonlinear model.

Since the structure is assumed known, a multi-dimensional nonlinear optimisation approach based on a Quasi-Newton method with a mixed quadratic and cubic line search procedure implemented by the MATLAB function "`fminunc`" of the Optimisation Toolbox (Coleman, Branch & Grace, 1999) was used.

One makes use of the known property of Gaussian BLA being the cascaded linear dynamics of a nonlinear system (Dobrowiecki & Schoukens, 2002; Schoukens, Lataire et al., 2009; Pintelon & Schoukens, 2012); in this case of the WH system, $G_{\text{BLA, B}}$ is given by:

$$G_{\text{BLA, B}} = c\,G_1 G_2$$

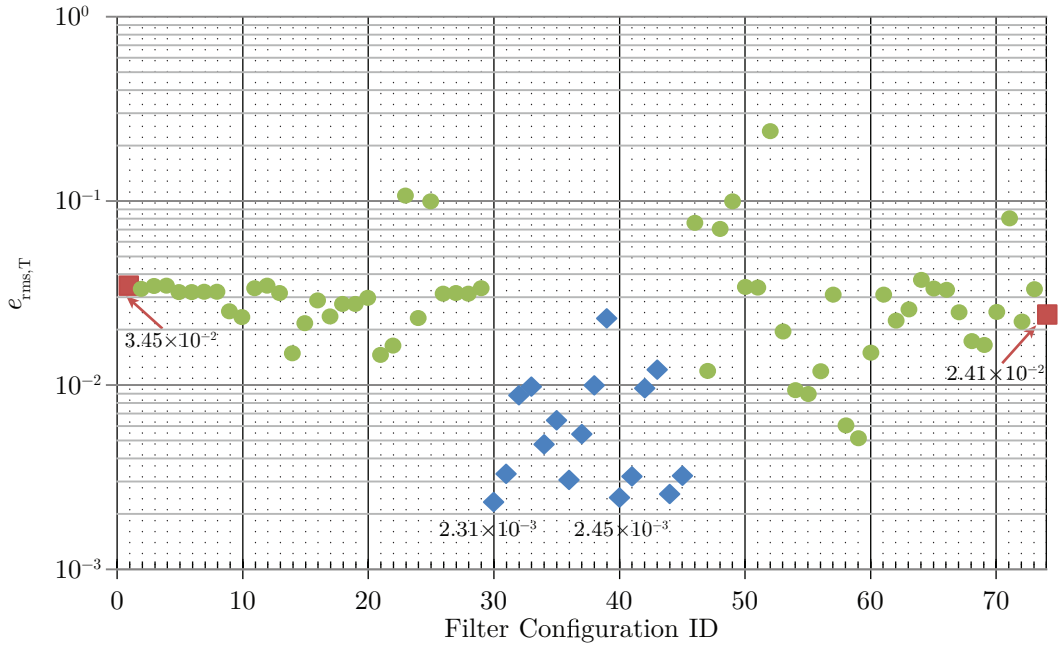
where G_1 is the frequency response function (FRF) of the first linearity, G_2 is that of the second linearity and c is some constant. By using (7.3.1), it is possible to use a brute-force approach to factorise the polynomials of the numerator and denominator of $G_{\text{BLA, B}}$ into zeros and poles respectively, and use an exhaustive search method for all possible combinations of assigning the zeros and poles into G_1 and G_2 while conforming to the conditions of realisable and causal (but not necessarily invertible) filters (Piegat, 2001, p. 528). This not only reduces the number of combinations to test for, it also allows the physical interpretation of the system to be retained.

Table 7.1 shows all possible combinations of distributing the zeros and poles to the first linearity G_1 and the second linearity G_2. As mentioned in the previous paragraph, the condition for realisability was imposed. This means that the transfer functions of G_1 and G_2 must be proper, i.e. there must be equal or fewer zeros than poles (for infinite impulse response (IIR) filters) unless there are no poles (resulting in finite impulse response (FIR) filters). This greatly reduces the number of possible combinations from $2^8 = 256$ to 74. In the table, '1' denotes the zero or pole being assigned to G_1 and '2' for G_2. Combinations with identification number (*ID*) 30 to 45 inclusive (see shaded *ID*s in the table) have zeros and poles of order 3 in both linear blocks, and these shall be labelled as 3/3 balanced filter configurations. Real zeros are denoted by *z1* and *z2* while complex conjugate pairs of zeros are denoted by *z34* and *z56*. Similar notation applies to the poles prefixed with letter *p*.

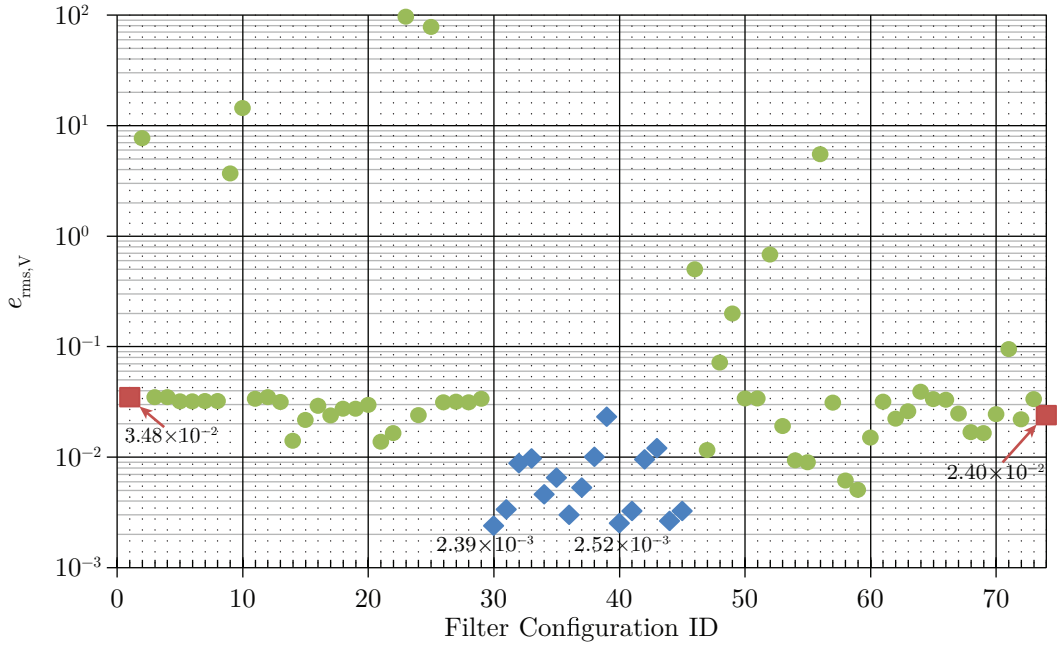
The training set and validation set rms errors of $e_{\text{rms,T}}$ and $e_{\text{rms,V}}$, respectively, are shown on a logarithmic scale in Fig. 7.3a and Fig. 7.3b, for all 74 zero-pole allocations. The lowest and second lowest values are indicated, as well as the values for the Wiener model ($G_1 = 1$) and the Hammerstein model ($G_2 = 1$). It can be

Table 7.1: Table of allocation of zeros and poles, '1' denotes assignment to G_1 and '2' for G_2. Shaded *ID*s emphasise symmetric allocations.

ID	Zeros				Poles				*ID*	Zeros				Poles			
	z1	*z2*	*z34*	*z56*	*p1*	*p2*	*p34*	*p56*		*z1*	*z2*	*z34*	*z56*	*p1*	*p2*	*p34*	*p56*
1	1	1	1	1	1	1	1	1	**38**	1	2	2	1	1	2	1	2
2	1	1	1	1	2	2	2	2	**39**	1	2	2	1	2	1	1	2
3	1	2	1	1	1	1	1	1	**40**	1	2	2	1	1	2	2	1
4	2	1	1	1	1	1	1	1	**41**	1	2	2	1	2	1	2	1
5	1	2	1	1	1	2	1	1	**42**	2	1	2	1	1	2	1	2
6	1	2	1	1	2	1	1	1	**43**	2	1	2	1	2	1	1	2
7	2	1	1	1	1	2	1	1	**44**	2	1	2	1	1	2	2	1
8	2	1	1	1	2	1	1	1	**45**	2	1	2	1	2	1	2	1
9	1	2	1	1	2	2	2	2	**46**	1	2	1	2	2	2	2	2
10	2	1	1	1	2	2	2	2	**47**	2	1	1	2	2	2	2	2
11	1	1	1	2	1	1	1	1	**48**	1	2	2	1	2	2	2	2
12	2	2	1	1	1	1	1	1	**49**	2	1	2	1	2	2	2	2
13	1	1	2	1	1	1	1	1	**50**	1	1	2	2	1	1	1	1
14	1	1	1	2	1	1	1	2	**51**	2	2	1	2	1	1	1	1
15	1	1	1	2	1	1	2	1	**52**	2	2	2	1	1	1	1	1
16	1	1	1	2	2	2	1	1	**53**	1	1	2	2	1	1	2	2
17	1	1	2	1	2	2	1	1	**54**	1	1	2	2	2	2	1	2
18	2	2	1	1	1	1	1	2	**55**	1	1	2	2	2	2	2	1
19	2	2	1	1	1	1	2	1	**56**	2	2	1	2	1	1	2	2
20	2	2	1	1	2	2	1	1	**57**	2	2	2	1	1	1	2	2
21	1	1	2	1	1	1	1	2	**58**	2	2	1	2	2	2	1	2
22	1	1	2	1	1	1	2	1	**59**	2	2	1	2	2	2	2	1
23	1	1	1	2	2	2	2	2	**60**	2	2	2	1	2	2	1	2
24	2	2	1	1	2	2	2	2	**61**	2	2	2	1	2	2	2	1
25	1	1	2	1	2	2	2	2	**62**	1	1	2	2	2	2	2	2
26	1	2	1	2	1	1	1	1	**63**	2	2	1	2	2	2	2	2
27	2	1	1	2	1	1	1	1	**64**	2	2	2	1	2	2	2	2
28	1	2	2	1	1	1	1	1	**65**	1	2	2	2	1	1	1	1
29	2	1	2	1	1	1	1	1	**66**	2	1	2	2	1	1	1	1
30	1	2	1	2	1	2	1	2	**67**	1	2	2	2	1	2	2	2
31	1	2	1	2	2	1	1	2	**68**	1	2	2	2	2	1	2	2
32	1	2	1	2	1	2	2	1	**69**	2	1	2	2	1	2	2	2
33	1	2	1	2	2	1	2	1	**70**	2	1	2	2	2	1	2	2
34	2	1	1	2	1	2	1	2	**71**	1	2	2	2	2	2	2	2
35	2	1	1	2	2	1	1	2	**72**	2	1	2	2	2	2	2	2
36	2	1	1	2	1	2	2	1	**73**	2	2	2	2	1	1	1	1
37	2	1	1	2	2	1	2	1	**74**	2	2	2	2	2	2	2	2



(a) Training set



(b) Validation set

Figure 7.3: Rms errors from running optimisation (up to degree 8) on all realisable zero-pole allocations in a Wiener-Hammerstein structure.
Diamonds: balanced order; Squares: Wiener or Hammerstein; Circles: others.

seen that the eight lowest errors are all given by 3/3 balanced filter configurations where both G_1 and G_2 are of order 3. This may suggest that the true system is of order 3/3, and indeed this is the case.

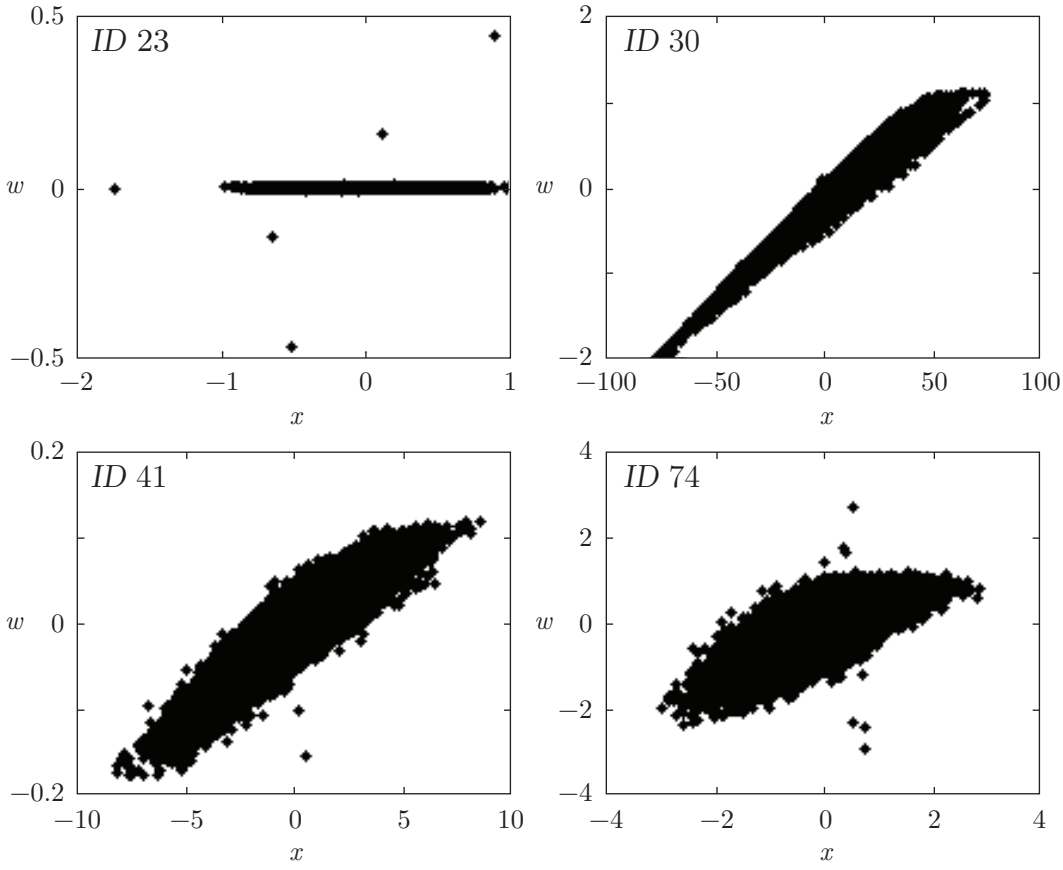


Figure 7.4: Graphs of nonlinearity output w versus input x

In practice, it is not necessary to run the optimisation procedure for all the zero-pole allocations as this is time-consuming. To alleviate this problem, the suitability of a particular combination can be gauged by plotting a graph of x versus w, and judging its potential based on the similarity of the plot to a static nonlinear curve. This is illustrated in Fig. 7.4, where four combinations are plotted; these are models with *ID 23* (very large error), *ID 30* (lowest error), and *ID 41* (relatively low error), and the Hammerstein model *ID 74* (moderate error). From Fig. 7.4, it is clear that the model with *ID* number *30* has a shape most closely resembling a nonlinear curve followed by models with *ID* numbers *41*, *74* and *23*, in that order. Only models having a shape similar to a static nonlinear curve need to be singled out for further optimisation. Note also that the large differences in scaling in x and w in the plots of Fig. 7.4 is not a point of concern because scaling by a constant

multiplier can be easily included later.

The lowest error values were achieved for one of the combinations with both G_1 and G_2 having model order 3; this is *ID* number 30 in Table 7.1 and Fig. 7.3. This can be written in the form:

$$\begin{aligned} G_1(z^{-1}) &= \frac{1 + b_1\, z^{-1} + b_2\, z^{-2} + b_3\, z^{-3}}{1 + a_1\, z^{-1} + a_2\, z^{-2} + a_3\, z^{-3}} \\ &= \frac{\left(1 - m_{\mathrm{r}1}\, z^{-1}\right)\, \left(1 - (m_{\mathrm{r}2} + \mathrm{j}m_{\mathrm{i}2})\, z^{-1}\right)\, \left(1 - (m_{\mathrm{r}2} - \mathrm{j}m_{\mathrm{i}2})\, z^{-1}\right)}{(1 - p_{\mathrm{r}1}\, z^{-1})\, (1 - (p_{\mathrm{r}2} + \mathrm{j}p_{\mathrm{i}2})\, z^{-1})\, (1 - (p_{\mathrm{r}2} - \mathrm{j}p_{\mathrm{i}2})\, z^{-1})} \end{aligned} \tag{7.7}$$

$$\begin{aligned} G_2(z^{-1}) &= \frac{1 + d_1\, z^{-1} + d_2\, z^{-2} + d_3\, z^{-3}}{1 + c_1\, z^{-1} + c_2\, z^{-2} + c_3\, z^{-3}} \\ &= \frac{\left(1 - m_{\mathrm{r}3}\, z^{-1}\right)\, \left(1 - (m_{\mathrm{r}4} + \mathrm{j}m_{\mathrm{i}4})\, z^{-1}\right)\, \left(1 - (m_{\mathrm{r}4} - \mathrm{j}m_{\mathrm{i}4})\, z^{-1}\right)}{(1 - p_{\mathrm{r}3}\, z^{-1})\, (1 - (p_{\mathrm{r}4} + \mathrm{j}p_{\mathrm{i}4})\, z^{-1})\, (1 - (p_{\mathrm{r}4} - \mathrm{j}p_{\mathrm{i}4})\, z^{-1})} \end{aligned} \tag{7.8}$$

where m and p are zeros and poles respectively, and the subscripts r and i represent real and imaginary parts, respectively. Since the static nonlinearity seems to be a smooth function, a polynomial

$$w = \mathfrak{f}(x) = \sum_{k=0}^{q} l_k x^k = \boldsymbol{l} \cdot \boldsymbol{x} \tag{7.9}$$

was used to describe the nonlinearity, where:

$$\boldsymbol{l} = \begin{bmatrix} l_0 & l_1 & \cdots & l_q \end{bmatrix}, \tag{7.10}$$

$$\boldsymbol{x} = \begin{bmatrix} x^0 & x^1 & \cdots & x^q \end{bmatrix}^{\mathrm{T}}. \tag{7.11}$$

The incremental nonlinear optimisation technique was used to solve for the parameters

$$\boldsymbol{\theta} = [m_{\mathrm{r}1} \quad m_{\mathrm{r}2} \quad m_{\mathrm{i}2} \quad m_{\mathrm{r}3} \quad m_{\mathrm{r}4} \quad m_{\mathrm{i}4} \quad p_{\mathrm{r}1} \quad p_{\mathrm{r}2} \quad p_{\mathrm{i}2} \quad p_{\mathrm{r}3} \quad p_{\mathrm{r}4} \quad p_{\mathrm{i}4}] \tag{7.12}$$

and $\boldsymbol{l}$, with the values obtained from ELiS, $\boldsymbol{\theta}_{\mathrm{ELiS}}$, as starting values for the iteration. The training set mean squared error defined in (7.1) was used as the cost function for "`fminunc`". In the first stage of the algorithm, G_1 was used to calculate x from u whereas G_2 was used to calculate w from y. A potential problem may arise if the zeros of G_2 were to fall outside the unit circle. In such a case, the filter inversion can be carried out in the frequency domain, so removing the constraint on the position of the zeros of G_2. The use of frequency domain based inversion to obtain w avoids the complication caused by the transmission zero, which is inherently difficult to invert—in the time domain, if there were at least one zero of G_2 outside the unit circle and y is used as input to the system $\left(G_2\right)^{-1}$, then in practice the output will diverge. If G_2 and $\left(G_2\right)^{-1}$ were both stable, there would be small differences between

the inversion obtained using the time domain and that obtained using the frequency domain, due to the fact that the latter assumes that steady-state is achieved and this does not actually happen. However, the problem can be alleviated by taking multiple periods of y before using Fourier inverse, and taking only the last period of the result as w.

The polynomial coefficients in $\boldsymbol{l}$ do not form part of the vector of parameters to be optimised with "`fminunc`", and instead, they are recalculated using the MATLAB built-in least squares regressor "`polyfit`" at the end of every iteration based on x and w (see (7.13) for the actual cost function used). This reduces the number of simultaneously optimised parameters to only 12. Initially, the polynomial degree was set to 1 (having a linear term and a constant term). The incremental approach proposed here gradually increases the polynomial degree. The main justification for not using a full degree polynomial initially is that such an approach may produce an optimisation surface which contains too many local minima. This is risky as it may cause the system to terminate at a local minimum rather than the global minimum—a common problem faced by many optimisation algorithms when the 'energy landscape' is complicated (Fallah-Jamshidi, Amiri & Karimi, 2010; Klock & Buhmann, 2000). One possible solution is to initiate the optimisation using several different sets of starting values (Venna & Kaski, 2006). This is, however, time-consuming and does not make optimal use of the results from the preceding investigation. Hence, a strategy is proposed here which is to gradually increment the degree of the polynomial. After obtaining the zeros and poles using "`fminunc`" with a first degree polynomial, these zeros and poles were used as starting values for the next optimisation now using a second degree polynomial. The new estimates of zeros and poles were then further used as starting values for a subsequent optimisation now using a third degree polynomial. This was continued until no further improvement was observed (at ninth degree polynomial). A summary of the program is given in Algorithm 1.

The estimates of G_1 and G_2 at the end of the optimisation using a polynomial of degree $q = 8$ were:

$$G_1(z^{-1}) = -10^{-2} \times \frac{1 - 5.0789\,z^{-1} - 0.8011\,z^{-2} - 1.4299\,z^{-3}}{1 - 2.1896\,z^{-1} + 1.7904\,z^{-2} - 0.5278\,z^{-3}}, \tag{7.14}$$

$$G_2(z^{-1}) = \frac{1 - 1.1371\,z^{-1} + 0.4170\,z^{-2} + 0.3818\,z^{-3}}{1 - 2.5953\,z^{-1} + 2.2718\,z^{-2} - 0.6691\,z^{-3}}. \tag{7.15}$$

The factor of 10^{-2} in G_1 was introduced to avoid poor numerical conditioning of the MATLAB "`polyfit`" routine. This scaling results in the range of x and w being

Algorithm 1 Parameter optimisation program

for $q = 1{+}{+}$ **do**
  if $q = 1$ **then**
    set initial guess $\boldsymbol{\theta}_1 = \boldsymbol{\theta}_{\text{ELiS}}$
  else
    set initial guess $\boldsymbol{\theta}_k = \boldsymbol{\theta}_{q-1}$
  end if
  find

$$\boldsymbol{\theta}_k = \underset{\boldsymbol{\theta},\ \texttt{fminunc}}{\arg\min} \left[e_{\text{rms,T}}(\boldsymbol{\theta}, \boldsymbol{l}')\right]\Big|_{\boldsymbol{l}' = \underset{\boldsymbol{l},\ \texttt{polyfit}}{\arg\min} |w - \boldsymbol{l} \cdot \boldsymbol{x}|^2} \tag{7.13}$$

  where the cost function for `fminunc`, is $e_{\text{rms,T}}$, as defined in (7.1).
  if $e_{\text{rms,T}}(\boldsymbol{\theta}_{q-1}) - e_{\text{rms,T}}(\boldsymbol{\theta}_q) > 5 \times 10^{-5}$ **then**
    $q \longrightarrow q + 1$
  else
    $\boldsymbol{\theta}_{\text{end}} \longleftarrow \boldsymbol{\theta}_{q-1}$
    break loop
  end if
end for

similar (see *ID 30* of Fig. 7.4). The static nonlinearity coefficients in l of (7.9) were:

$$\begin{aligned}\boldsymbol{l} = 10^{-3} \times [&0.0223 \quad 12.2655 \quad -0.5930 \quad -5.1214 \\ &-4.3824 \quad 2.3266 \quad 2.5911 \quad -0.3599 \quad -0.4274]\end{aligned} \tag{7.16}$$

Table 7.2: Performance measures using the incremental nonlinear optimisation technique in terms of polynomial degree q

q	μ_{T}	μ_{V}	$e_{\text{rms,T}}$	$e_{\text{rms,V}}$
1	3.32×10^{-4}	1.31×10^{-4}	4.22×10^{-2}	4.32×10^{-2}
2	4.29×10^{-5}	-2.71×10^{-4}	2.34×10^{-2}	2.33×10^{-2}
3	-6.61×10^{-5}	-7.36×10^{-4}	1.56×10^{-2}	1.51×10^{-2}
4	-1.21×10^{-4}	-2.90×10^{-4}	8.83×10^{-3}	8.73×10^{-3}
5	-1.29×10^{-4}	-1.56×10^{-4}	6.06×10^{-3}	6.09×10^{-3}
6	-8.48×10^{-5}	-1.54×10^{-4}	4.33×10^{-3}	4.22×10^{-3}
7	-8.52×10^{-5}	-1.08×10^{-4}	3.76×10^{-3}	3.78×10^{-3}
8	-3.96×10^{-5}	-3.81×10^{-5}	2.35×10^{-3}	2.43×10^{-3}

The performance measures μ_{T}, μ_{V}, $e_{\text{rms,T}}$ and $e_{\text{rms,V}}$ are shown in Table 7.2. Values of σ_{T} and σ_{V} are omitted since they are the same as $e_{\text{rms,T}}$ and $e_{\text{rms,V}}$, respectively, up to 3 significant figures; this is due to the small magnitude of the mean errors μ_{T} and μ_{V}. For polynomial degree of 8, there was a reduction in $e_{\text{rms,V}}$

by a factor of 23.74 compared with the value of 5.77×10^{-2} obtained using the linear model based on the BLA of $G_{\mathrm{BLA,B}}$. The performance measures of a similar model optimised directly without the incremental polynomial degree technique were also calculated. The values were $\mu_{\mathrm{T}} = -8.59\times10^{-4}$, $\mu_{\mathrm{V}} = -3.07\times10^{-4}$, $\sigma_{\mathrm{T}} = 6.82\times10^{-3}$, $\sigma_{\mathrm{V}} = 6.75\times10^{-3}$, $e_{\mathrm{rms,T}} = 6.82\times10^{-3}$ and $e_{\mathrm{rms,V}} = 6.76\times10^{-3}$; both $e_{\mathrm{rms,T}}$ and $e_{\mathrm{rms,V}}$ were approximately a factor of three greater than those in Table 7.2 for polynomial degree 8, illustrating the merits of the incremental technique.

With respect to initialising the poles and zeros from $G_{\mathrm{BLA,A}}$, while the convergence was quicker, applying the incremental nonlinear optimisation technique led to poorer results compared with that when initialising from $G_{\mathrm{BLA,B}}$. The method converged at a polynomial of degree 3, giving $\mu_{\mathrm{T}} = -1.39\times10^{-4}$, $\mu_{\mathrm{V}} = -2.62\times10^{-4}$, $\sigma_{\mathrm{T}} = 8.96\times10^{-3}$, $\sigma_{\mathrm{V}} = 8.80\times10^{-3}$, $e_{\mathrm{rms,T}} = 8.97\times10^{-3}$ and $e_{\mathrm{rms,V}} = 8.81\times10^{-3}$. Upon closer inspection, it was found that the inferior results for $G_{\mathrm{BLA,A}}$ were due to the slightly incorrect shaping of the notch in $|G_2|$; see Fig. 7.5b for the plot of the true $|G_2|$. An inaccurate initial estimate led to subsequent degradation of the estimates of the zeros associated with this notch. Even though the overall fit improved with the incremental nonlinear technique, the error in the positioning of this notch caused the final results to be worse than those obtained by starting with $G_{\mathrm{BLA,B}}$. For the latter, the problem did not occur as $G_{\mathrm{BLA,B}}$ was estimated using harmonics 1 to 8000 only (see Section 7.3), thereby excluding the portion where the notch in $|G_2|$ occurs. As a consequence, $G_{\mathrm{BLA,B}}$ was almost flat across the region beyond harmonic 8000. The notch only formed when the degree of the polynomial was increased to 4.

7.4 Simultaneous parameter optimisation

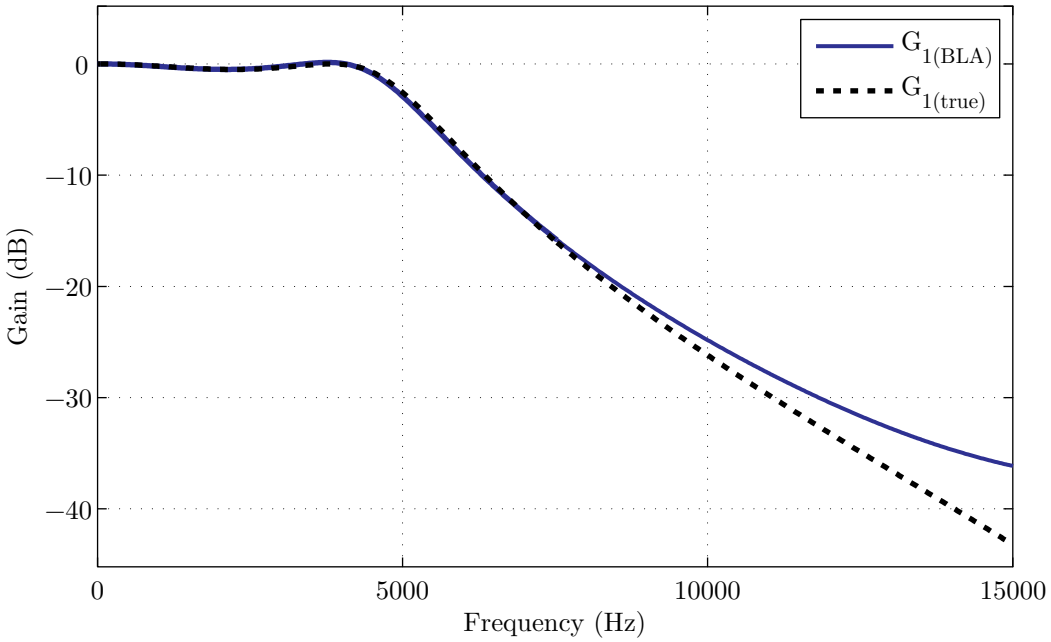
With the model obtained using the technique described in Section 7.3.1, simultaneous parameter optimisation was subsequently carried out. All the model parameters

$$\boldsymbol{\gamma} = [\boldsymbol{\theta} \quad \boldsymbol{l}] \tag{7.17}$$

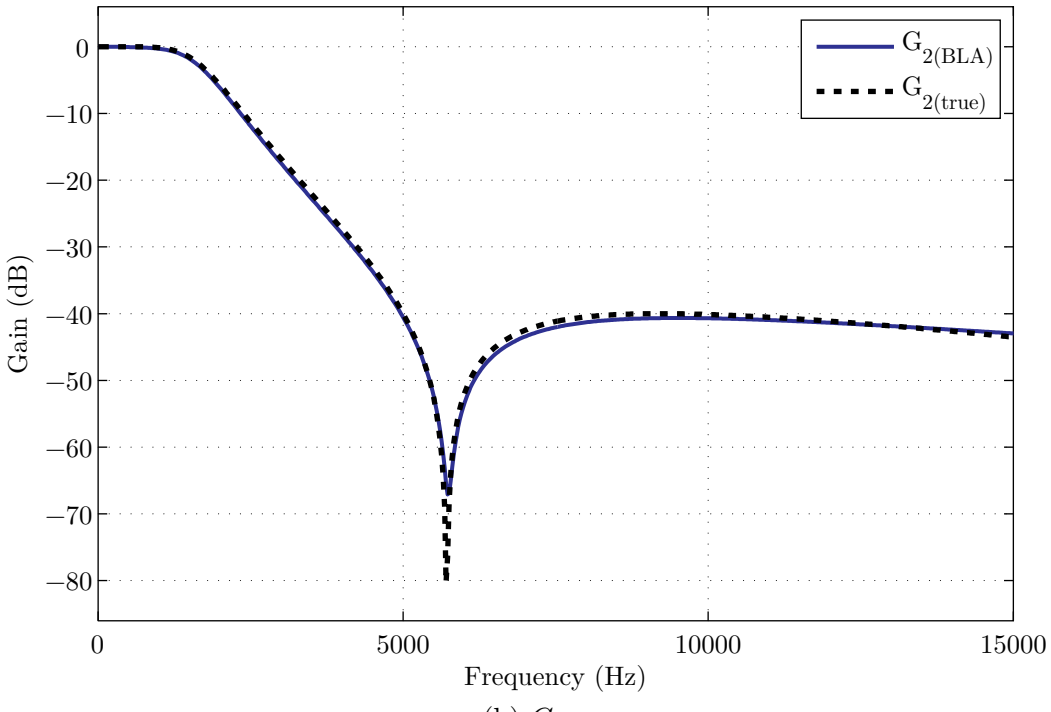
were now re-optimised simultaneously using "`fminunc`", to solve for

$$\boldsymbol{\gamma}_q = \arg\min_{\boldsymbol{\gamma}} \left[e_{\mathrm{rms,T}}(\boldsymbol{\gamma})\right] \tag{7.18}$$

This step did not require any filter inversion. Models having polynomial degrees ranging from 8 to 12 were tried. The best results were obtained using the model with polynomial of degree 10. Note that the incremental technique was not applied



(a) G_1



(b) G_2

Figure 7.5: Comparison between the gain responses of the estimated transfer functions (a) G_1 and (b) G_2 with theoretical values.

here as $\boldsymbol{\gamma}$ was being optimised rather than $\boldsymbol{\theta}$. In particular, if the incremental technique were utilised, $\boldsymbol{l} = [l_0 \quad l_1 \quad \cdots \quad l_q]$ may not converge.

The estimates of G_1 and G_2 after applying the simultaneous parameter optimisation were:

$$G_1(z^{-1}) = -10^{-2} \times \frac{1 - 4.9987\,z^{-1} - 1.1937\,z^{-2} - 1.5613\,z^{-3}}{1 - 2.1863\,z^{-1} + 1.7850\,z^{-2} - 0.5251\,z^{-3}}, \tag{7.19}$$

$$G_2(z^{-1}) = \frac{1 - 1.1979\,z^{-1} + 0.5077\,z^{-2} + 0.3133\,z^{-3}}{1 - 2.5979\,z^{-1} + 2.2765\,z^{-2} - 0.6712\,z^{-3}}. \tag{7.20}$$

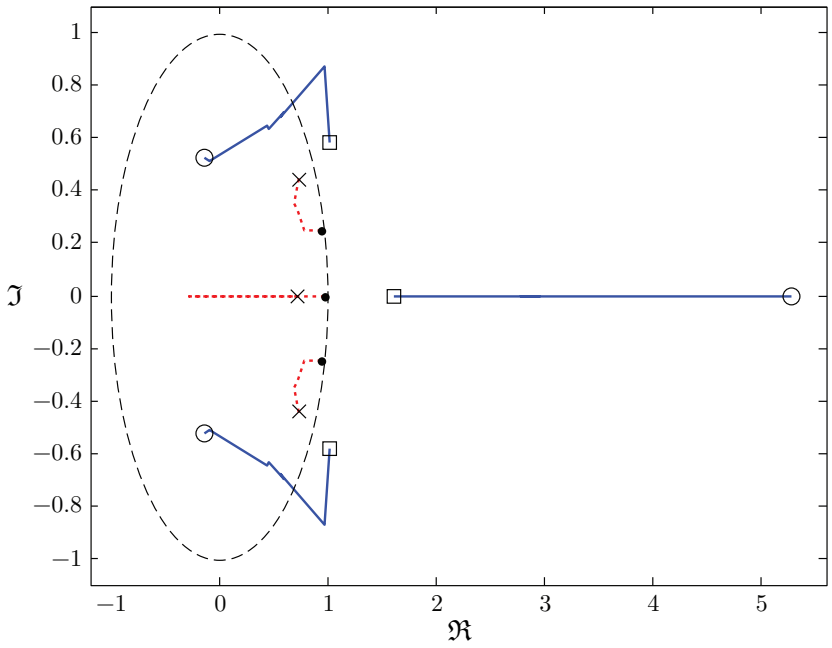
The static nonlinearity coefficients in l of (7.9) were:

$$\begin{aligned}\boldsymbol{l} = 10^{-3} \times [&0.0025 \quad 12.1710 \quad 0.1077 \quad -4.7597 \quad -6.1076 \\ &2.3276 \quad 4.7434 \quad -0.5441 \quad -1.4882 \quad 0.0478 \quad 0.1528]\end{aligned} \tag{7.21}$$

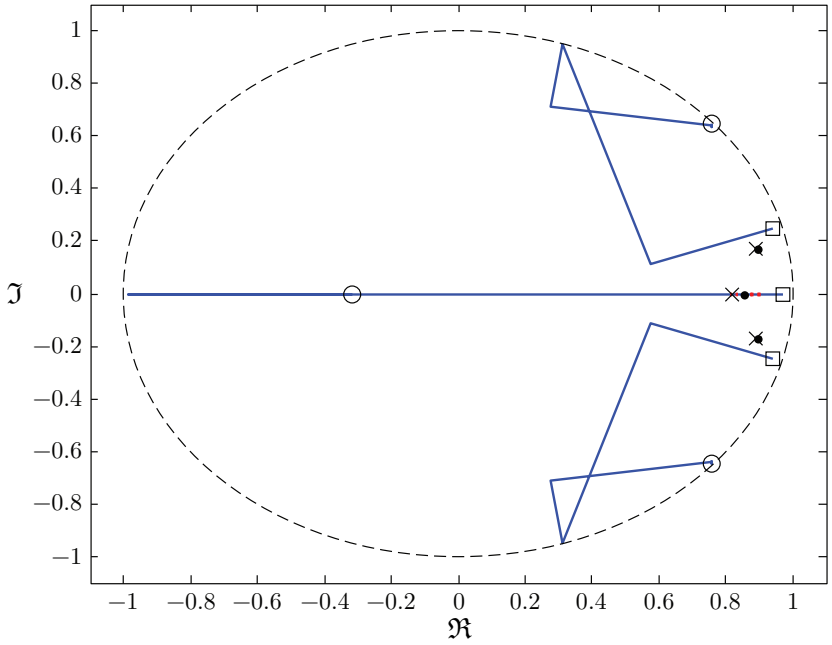
The performance measures were $\mu_{\mathrm{T}} = -4.45{\times}10^{-6}$, $\mu_{\mathrm{V}} = -5.55{\times}10^{-6}$, $\sigma_{\mathrm{T}} = e_{\mathrm{rms,T}} = 1.54{\times}10^{-3}$ and $\sigma_{\mathrm{V}} = e_{\mathrm{rms,V}} = 1.55{\times}10^{-3}$. The value of $e_{\mathrm{rms,V}}$ was reduced by a factor of 37.23 compared with that obtained using the linear model based on $G_{\mathrm{BLA,B}}$.

Although not asked for in the *Benchmark* specification, it is interesting to compare the gain responses of the estimated transfer functions G_1 and G_2 with the theoretical values; this is shown in Fig. 7.5. Based solely on the observation of u and y, it is only possible to determine a gain for the overall system, rather than that of the individual blocks of G_1 and G_2 separately. For the purposes of comparison, the dc gain of both linear blocks has been set to unity, which is the same as that of the corresponding theoretical filter. For G_2, a very close match was obtained, whereas for G_1, deviation occurs only when the frequency is close to the cut-off frequency of the Gaussian excitation signal at 10 kHz. It is expected that systems with nonlinearity cannot be easily extrapolated beyond the range covered by data from the training set. The good matching below 10 kHz shows that the technique described in this chapter is effective in separating the dynamics of the two linear sub-systems in a WH system.

The evolution of zeros and poles from the values obtained from (7.6) to those corresponding to (7.19) and (7.20) is shown in Fig. 7.6. It can be seen that, with the exception of the poles of G_2, there is a change, sometimes substantial, in the positions of the zeros and poles. This shows that the estimates were free to evolve without being stuck in local minima—supporting the case for a convergence to the global minimum. As for the lack of changes seen for the poles of G_2, it is likely that the poles were estimated accurately from the start by ELiS due to them representing low frequency components, and thus were less affected by the exclusion of higher



(a) G_1



(b) G_2

Figure 7.6: Evolution of zeros (blue solid lines) and poles (heavy red dotted lines) for (a) G_1 and (b) G_2. The unit circle is marked with dashed lines. The final values that correspond to (7.19) and (7.20) are marked with open circles for zeros and crosses for poles. The initial values which correspond to (7.6) are marked with open squares for zeros and dots for poles.

frequencies during the estimation using ELiS (see Section 7.3).

7.5 Fine-tuning of static polynomial coefficients

While the method applied in Section 7.4 was able to reduce the error from the value achieved in Section 7.3.1, it was found that the routine in Section 7.4 is more time-consuming. The fact that no further improvement was obtained when using a polynomial of degree higher than 10 indicates that the method is unable to cope with the increasing number of parameters to be optimised. Hence, at this point, using the estimates of G_1 and G_2 in (7.19) and (7.20), x was re-calculated from u using G_1 whereas w was re-calculated from y using $\left(G_2\right)^{-1}$.

With these values of x and w, a polynomial was fitted using linear least squares. The best degree of polynomial was decided based on the smallest value of $e_{\mathrm{rms,T}}$, since it is specified in the *Benchmark* exercise that the validation data should not be used for any purpose during the tuning of the model, including for model selection. The results are shown in Fig. 7.7, and the best degree is 31, although there is comparatively little difference in $e_{\mathrm{rms,T}}$ for degrees above 32. It is important to note that, in theory, increasing the degree of polynomial indefinitely should result in a monotonic decrease in the *training* set error $e_{\mathrm{rms,T}}$. Linear least squares regression is a linear procedure and therefore there is no danger of being stuck in local minima. The reason that there is a slight rise in the training set error past degree 31 was purely due to numerical condition problems with the "`polyfit`" routine in MATLAB when the data matrix was large and the polynomial degree was high. For $e_{\mathrm{rms,V}}$, the minimum coincides with degree 32. Beyond degree 33, $e_{\mathrm{rms,V}}$ increases rapidly due to a combination of errors introduced by numerical conditioning problems and over-modelling. Over-modelling refers to the fact that the estimated parameters are increasingly being influenced by noise present in the training set, which leads to a decreasing model quality for the validation set. Using degree 31, the performance measures were $\mu_{\mathrm{T}} = -1.53\times10^{-5}$, $\mu_{\mathrm{V}} = 6.63\times10^{-6}$, $\sigma_{\mathrm{T}} = e_{\mathrm{rms,T}} = 7.18\times10^{-4}$ and $\sigma_{\mathrm{V}} = e_{\mathrm{rms,V}} = 6.94\times10^{-4}$.

By plotting w versus x in a similar manner as in Fig. 7.4, it was found that the static nonlinearity characteristic had two distinct regions. The least squares procedure was repeated, using a dual-polynomial of equal order in both regions, and with various values of breakpoints between the regions. The lowest value of $e_{\mathrm{rms,T}}$ was obtained with a breakpoint of $x = 0.34$. The simulation error results are shown in Fig. 7.8, from which it can be observed that the best degree of dual-polynomial has decreased to degree 5. This is not surprising, because a 'breakpoint'

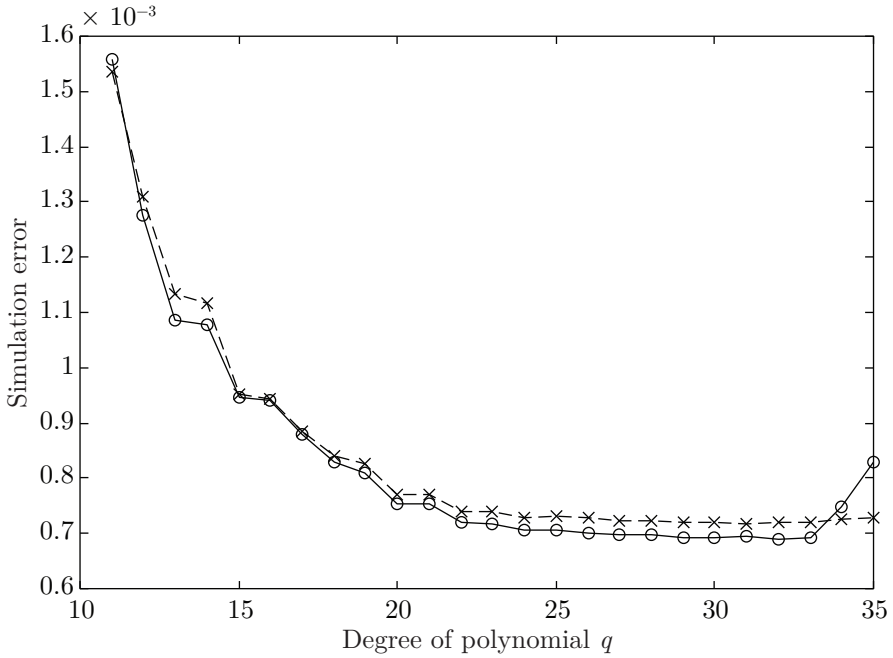


Figure 7.7: Effect of polynomial degree on simulation error for the training dataset (dashed line with crosses) and the validation dataset (solid line with circles)

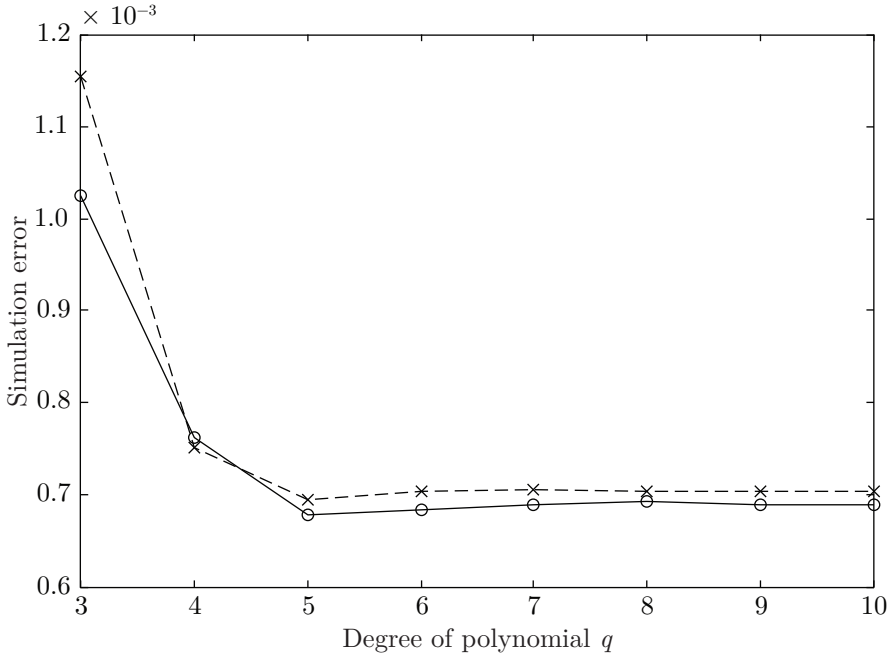


Figure 7.8: Effect of dual-polynomial degree on simulation error for the training dataset (dashed line with crosses) and the validation dataset (solid line with circles)

is able to capture a high degree of nonlinearity. Thus, the number of parameters for modelling the nonlinearity is reduced from 32 (including the constant value) to 13 (including the two constant values and the breakpoint).

The final static nonlinearity was described by the dual-polynomial:

$$\boldsymbol{l} = \begin{cases} [-0.0095 \quad 11.7464 \quad -0.1086 \\ \quad -0.4413 \quad -0.3859 \quad -0.1111] & \text{for } x \leq 0.34 \\ [-4.5403 \quad 41.5374 \quad -64.0957 \\ \quad 50.2411 \quad -19.2945 \quad 2.8837] & \text{for } x > 0.34 \end{cases} \tag{7.22}$$

The performance measures were $\mu_{\mathrm{T}} = 4.56{\times}10^{-7}$, $\mu_{\mathrm{V}} = 2.02{\times}10^{-5}$, $\sigma_{\mathrm{T}} = e_{\mathrm{rms,T}} = 6.95{\times}10^{-4}$, $\sigma_{\mathrm{V}} = 6.78{\times}10^{-4}$ and $e_{\mathrm{rms,V}} = 6.79{\times}10^{-4}$. The value of $e_{\mathrm{rms,V}}$ was reduced by a factor of 84.98 compared with that obtained using the linear model based on $G_{\mathrm{BLA,B}}$.

The model output and the simulation error are plotted in Fig. 7.9. The actual output is not shown, because it practically overlaps with the model output. It can be seen from Fig. 7.9 that the error is indeed very small, in comparison with the model output. The discrete Fourier transform (DFT) spectra are shown in Fig. 7.10.

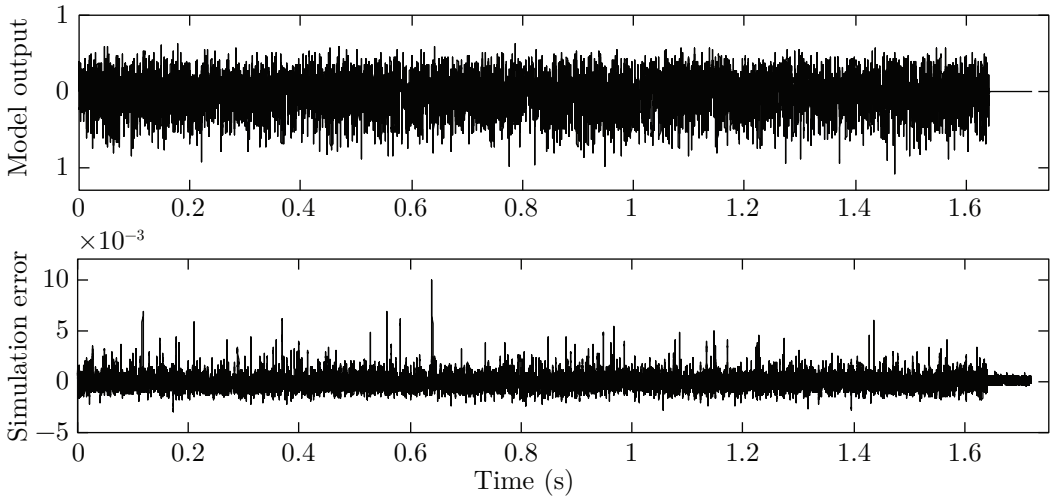


Figure 7.9: Model output and simulation error for the training data set

Again not part of the *Benchmark* specification, it is interesting to compare the fit of the normalised nonlinear characteristic, obtained using the linear least squares regressor function "`polyfit`" in MATLAB, with the characteristic expected from the circuit shown in Schoukens, Suykens and Ljung (2009, fig. 2). The circuit consists of a potential divider of 1 kΩ and 10 kΩ, with the output taken across the

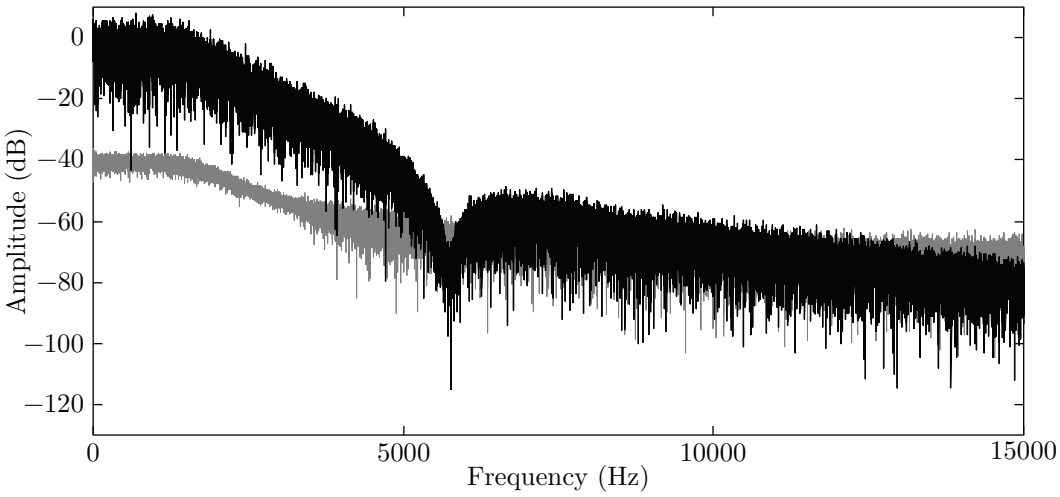


Figure 7.10: DFT spectra of the model output (black) and simulation error (grey) for the training data set

10 kΩ resistor and a *1N4148* diode in parallel. As noted by Wills and Ninness (2012, eq. 50), in the 0.1 mA to 100 mA range (at 25 °C), the voltage-current relationship of a typical *1N4148* diode can be described by:

$$\log_{10} i_{\text{diode}} = 9.38 v_{diode} - 8.69 \tag{7.23}$$

Using Kirchhoff's laws for the circuit and the voltage-current relationship in (7.23), the nonlinear relationship between the input voltage v_i and the output voltage v_o can be expressed by:

$$v_\text{i} = \frac{11}{10} v_\text{o} + 10^{(9.38 v_\text{o} - 5.69)} \tag{7.24}$$

For negative output voltages, the second term on the right-hand side of (7.24) is negligible, so that $v_\text{o}/v_\text{i} \approx 10/11$. For $v_\text{o} > 5.69/9.38 = 0.607\,\text{V}$, the bracket in the second term becomes positive, and this term begins to dominate, resulting in a much smaller slope of v_o/v_i.

The intermediate signals x ($= v_\text{i}$) and w ($= v_\text{o}$), estimated using filter configuration *ID 30* are plotted (with the vertical axis scaled to match theoretical characteristic) in Fig. 7.11. It may be seen that there is close agreement between estimated values and the theoretical characteristic throughout the range of the input voltage x. The problem whereby a high-order polynomial rapidly diverges even slightly beyond the range of the input is seen to be considerably reduced here when using the lower order dual-polynomial.

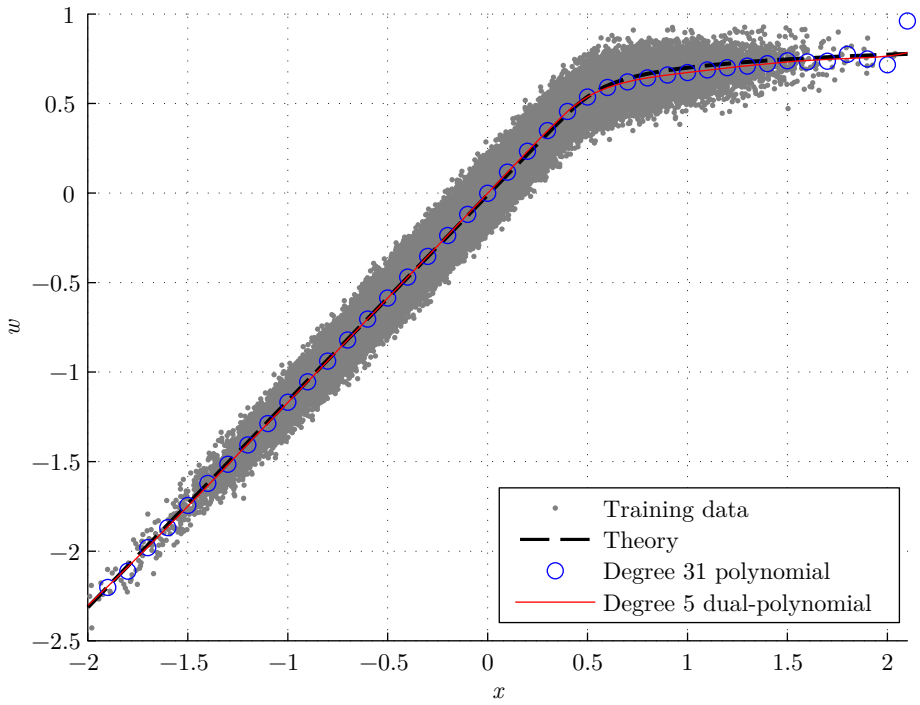


Figure 7.11: Static nonlinearity characteristic using coefficients from filter *ID 30*

7.6 Summary of the proposed approach and comparison with other approaches

A summary of the main steps in the proposed approach is given below:

Table 7.3: Computational time required

Step	Sub-systems optimised	Approximate computational time
1	Linear	$<$ 5 seconds
2	Linear	$<$ 1 second per combination
3	Linear	$<$ 1–3 minutes per degree of polynomial
4	Linear and nonlinear	15–20 minutes per degree of polynomial
5	Nonlinear	$<$ 5 seconds per degree of polynomial

Table 7.3 shows the approximate computational time required for each step when the algorithms were executed on a personal computer with Intel Core 2 Duo E7400 processor running at 2.8 GHz with 2 GiB of memory.

A comparison between the performance of the model obtained using the proposed technique with that from other models applied to the same set of data is shown in Table 7.4, from which it is clear that the training data error of the

Algorithm 2 Summary of the iterative nonlinear optimisation procedure on a Wiener-Hammerstein (WH) system

Step 1: Identification of the Best Linear Approximation (BLA)
set $\boldsymbol{l} = [0\ 1]$
optimise $G_{\mathrm{BLA}}(z^{-1})$
Step 2: Zero-pole allocation via nonlinear curve plotting
for all combinations of realisable zero-pole allocation **do**
plot(x, w)
if the plot resembles a static nonlinear curve (user-decision) **then**
include filter combination (ID) into set $\boldsymbol{X}$ for further test
end if
end for
Step 3: Incremental nonlinear optimisation
for all *ID*s $\in \boldsymbol{X}$ **do**
optimise $\boldsymbol{\theta}$ and $\boldsymbol{l}$ using **Algorithm 1**
end for
$\mathrm{ID}_{\mathrm{selected}} = \arg\min_{\mathrm{ID}} e_{\mathrm{rms,\ T}}$
Step 4: Simultaneous parameter optimisation
optimise $\boldsymbol{\gamma} = [\boldsymbol{\theta} \quad \boldsymbol{l}]$
Step 5: Fine-tuning of static polynomial coefficients
plot(x, w)
look for a breakpoint making use of visual information from the plot
optimise $\boldsymbol{l}$ for dual-polynomial static nonlinearity fitting

proposed model is excellent considering the small number of parameters used—25 in all, 12 for modelling the linear filters and 13 for modelling the nonlinearity.

7.7 Conclusions

In this chapter, a strategy was presented for fully identifying a Wiener-Hammerstein (WH) system, proposed in a *Benchmark* excercise in the 15th International Federation of Automatic Control (IFAC) Symposium on System Identification (SYSID) 2009 (Schoukens, Suykens & Ljung, 2009). The strategy utilised a property of the Gaussian Best Linear Approximation (BLA)—its ability to decouple the dynamics of the first and second linearities by splitting the poles and zeros contained in the BLA. The brute force approach of poles and zero assignment can be speeded up by imposing the dynamical linearities to be causal and realisable, and by visually examining the mapping between the input and output data of the static nonlinearity on a scatter diagram.

Table 7.4: Comparison between several models

Source	Model	$e_{\text{rms, V}}$	Parameters
Falck et al. (2012)	Least Squares Support Vector Machines (LS-SVM)	8.88×10^{-3}	30
Lopes dos Santos, Ramos and Martins de Carvalho (2012)	Hammerstein-bilinear	1.09×10^{-2}	90
Paduart, Lauwers, Pintelon and Schoukens (2012)	Polynomial Nonlinear State-space (PNLSS)	4.18×10^{-4}	797
Piroddi, Farina and Lovera (2012)	Nonlinear Auto-Regressive eXogenous (NARX)	1.64×10^{-2}	15
Sjöberg, Lauwers and Schoukens (2012)	Wiener-Hammerstein	3.0×10^{-4}	64
Truong and Wang (2009)	Wavelet based state-dependent parameter	1.45×10^{-2}	23
Wills and Ninness (2012)	Generalised Hammerstein-Wiener	4.81×10^{-4}	30
Tan, Wong and Godfrey (2012)	Wiener-Hammerstein	6.79×10^{-4}	25

With the poles and zeros assigned to the two dynamical linearities, their values were used as initial seeds for an incremental nonlinear optimisation approach. The strategy adopted had several stages. First, the parameters of a linear model were optimised via a sequence of steps, then the all parameters were optimised simultaneously, and in the final stage, only the parameters of the nonlinear block were optimised. The final model was simple in terms of structure and had relatively few parameters. The strategy has worked well on the *Benchmark* data, with small error values being obtained for both the training data and the validation data. There was good agreement between the normalised gain responses of the estimated and theoretical filters; also, good agreement between the dual-polynomial fit of the nonlinearity and the theoretical nonlinear characteristic.

The approach taken would seem equally applicable to the identification of other systems with a WH structure provided the nonlinearity is static (no hysteresis) and an appropriate basis function is chosen to model the nonlinearity.

Chapter 8

Benchmark Study: A Greybox Approach to Modelling a Hyperfast Switching Peltier Cooling System

This chapter describes an approach to modelling a 'hyperfast' switching Peltier cooling system, which has been used as a benchmark for the identification and modelling of complex interconnected systems in the U.K. Automatic Control Council (Ukacc) International Conference on Control, 2010 (Control 2010) (Cham, Tan & Tan, 2010). The overall system consists of three subsystems. The first two (*Subsystems 1* and *2*) are connected in parallel, with their outputs both fed as inputs to the third (*Subsystem 3*) and as such, this is an multiple-input single-output (Miso) system.

Two sets of data were made available, both having multisines applied to the inputs of *Subsystems 1* and *2*. The first, called the training set, consisted of the two multisine inputs and the output signal from the *Subsystem 3*. The second, called the validation set, had the multisine inputs reversed and again had the output signal from *Subsystem 3*, but unlike the training set, also had the intermediate outputs from *Subsystems 1* and *2*.

The two multisines were both of period $N = 600$, and they both had their non-zero harmonics uniform in amplitude. *Multisine A* had harmonic components at frequency lines $1, 7, 13, 19, \ldots, 235$ only, while *Multisine B* had components at frequency lines $5, 11, 17, 23, \ldots, 239$ only. Note that both of these have harmonic multiples of 2 and 3 suppressed, which results in the effect of even-order nonlinear distortions being eliminated at the output, and in the effect of odd-order nonlinear

distortions being reduced (Pintelon and Schoukens, 2001; Tan, Godfrey and Barker, 2005). In the training set, *Multisine A* was applied to *Subsystem 1* and *Multisine B* was applied to *Subsystem 2*, this order was *reversed* in the validation set.

From the information given in Cham, Tan and Tan (2010) (hereinafter referred as 'the *Benchmark*'), *Subsystem 1* contained a highly nonlinear element (a stepper motor). However, the step time of the motor was only 0.05 s, so that there are 20 such steps within each sampling interval of 1 s. Further, with a movement of 7.5° per step, the motor can move 150° within each sampling interval, and the range of operation was confined to be from 75° to 225°. The relationship between angle and flow rate, given in Cham, Tan and Tan (2010, fig. 3) was nonlinear, but over the operating range was almost linear. Again from the information of the Benchmark, *Subsystem 2* was very nearly linear.

In the figures given in the *Benchmark*, there is almost no spectral content in the output at any harmonic other than those of either of the input signals, partly as a result of the harmonic suppression in the inputs, but also suggesting that the overall system was very nearly linear.

8.1 System modelling

In system identification, the term *whitebox* modelling refers to the process of modelling a system through first principles, laws of physics and explicit assumed relationships between the input and output through prior knowledge of the system, resulting in a *mechanistic* model. The term *blackbox* modelling refers to the process of modelling a system through non-parametric techniques, without knowledge of physical inner workings of the system, resulting in an *empirical* model. A '*greybox*' modelling approach is a mixture between whitebox and blackbox modelling.

A greybox approach was taken to model each of the three subsystems separately. Detailed information given in the *Benchmark* readily allowed *Subsystem 1* to be modelled as whitebox. *Subsystem 2* was modelled as blackbox with the knowledge given in the Benchmark of it being very linear in its characteristics. A trial-and-error approach was then taken to model *Subsystem 3* to condition the outputs from *Subsystems 1* and *2* to give the output frequency spectrum and time domain response as close as possible to those given in the Benchmark.

A block diagram of the overall system is shown in Fig. 8.1. The outputs of the single-input single-output (Siso) *Subsystems 1* and *2* are combined to form the inputs to the dual-input single-output *Subsystem 3*.

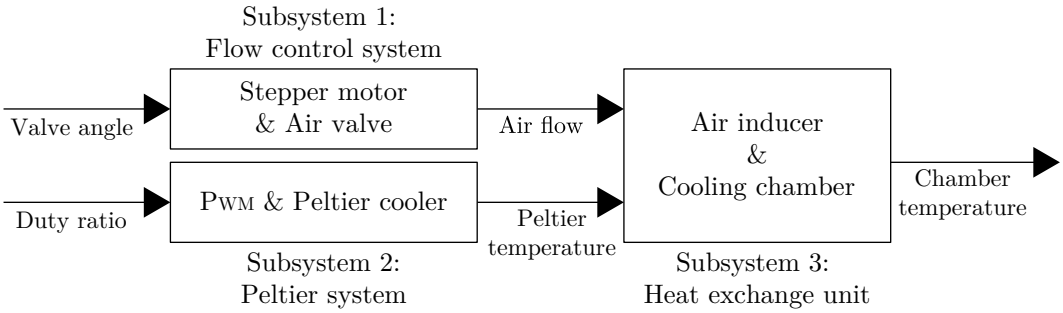
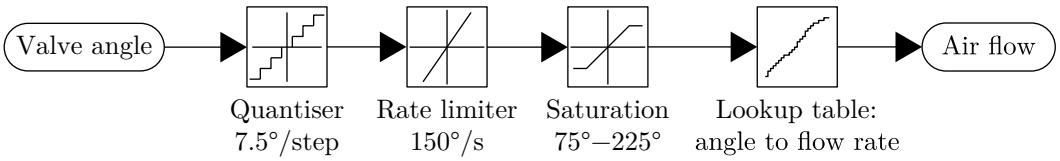


Figure 8.1: Overview of the Peltier cooling system

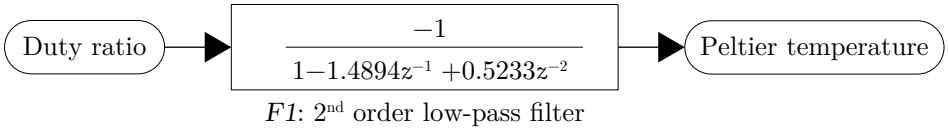
8.1.1 Flow control system

Subsystem 1 consisted of an air flow valve controlled by a stepper motor, with the input signal given to the controller of the motor being the desired valve angle. The input multisine signals used were scaled to span the input amplitude range, from 75° to 225°.

A MATLAB *Simulink* model utilising a rate limiter, quantisation nonlinearity, saturation nonlinearity and a lookup table translating valve angle to air flow rate was developed with aid of the steady-state gain characteristic (Cham, Tan & Tan, 2010, fig. 3) and specified stepper motor characteristics. This is shown in Fig. 8.2a.



(a) *Subsystem 1*: Flow control system



F1: 2nd order low-pass filter

(b) *Subsystem 2*: Peltier system

Figure 8.2: Block diagrams of *Subsystems 1* and *2*

8.1.2 Switching Peltier cooling system

Subsystem 2 was a Peltier cooler with the duty ratio of a pulse-width modulation (PWM) waveform used to vary the power supplied to a 120 W Peltier module. The input signal was scaled to span the input amplitude range from 0 to 1, which represents the duty ratio. The characteristics given in the *Benchmark* suggests that

this subsystem was very nearly linear and that it had much slower dynamics than *Subsystem 1*. The system dynamics were modelled by a simple second order digital low pass filter *F1*, as shown in Fig. 8.2b, to approximate the very nearly linear and low pass characteristics. The order of the filter was decided on by observing the simplicity of the transfer characteristics from the output of *Subsystem 2* to the overall system output with respect to excited frequency lines of input *Multisine B*. The gain of this subsystem was negative, since an increase in the duty ratio results in a decrease of temperature at the output. A guess was made to the filter coefficients by observing the harmonic components of the master output incorporating both *Subsystem 3 Path 2* and *Subsystem 2* initially. The coefficients were subsequently modified through a trial-and error approach to match the output spectrum of the intermediate signal for *Subsystem 2*.

8.1.3 Heat exchange unit

Subsystem 3 physically comprised a cool air inducer attached to a cooling chamber. From a modelling perspective the subsystem conditions and combines the signals from *Subsystems 1* and *2*. This system was modelled as a blackbox, and because of a lack of physical knowledge of *Subsystem 3*, the model structure had redundant gain blocks to allow for flexibility in the trial-and-error modelling process.

The model shown in Fig. 8.3 was constructed by comparing the spectrum of the overall system output (from *Subsystem 3*) in the training data with the input spectra of *Multisine A* for the *Subsystem 1* signal path and *Multisine B* for the *Subsystem 2* signal path to deduce appropriate filters and relative contributions between these two paths. All filter frequencies specified from this point onwards are normalised in the range of 0 to 1.

For the *Subsystem 1* pathway, the resulting structure was a combination of a fourth order finite impulse response (FIR) band stop filter *F2* with stop at about 200/600 and a second order high pass filter *F3* (cascaded by two first order Butterworth filters in series) with a low cut-off frequency of 1/600. *F2* was designed using arbitrary magnitude fitting function "`arbmag`" in "`fdesign`", and the GUI "`fdatool`" of the Signal Processing Toolbox of *Simulink* (Krauss et al., 1994) by observing the transfer characteristics from the output of *Subsystem 1* to the overall system output with respect to excited frequency lines of input *Multisine A* (Cham, Tan & Tan, 2010, eq. 1). The filter order of four was chosen on the criteria of simplicity and sufficiency, based on previews of frequency responses in the Toolbox. *F3*, designed using the same Toolbox with standard Butterworth filter coefficients, gave the appropriate attenuation seen on the first harmonic number of the training

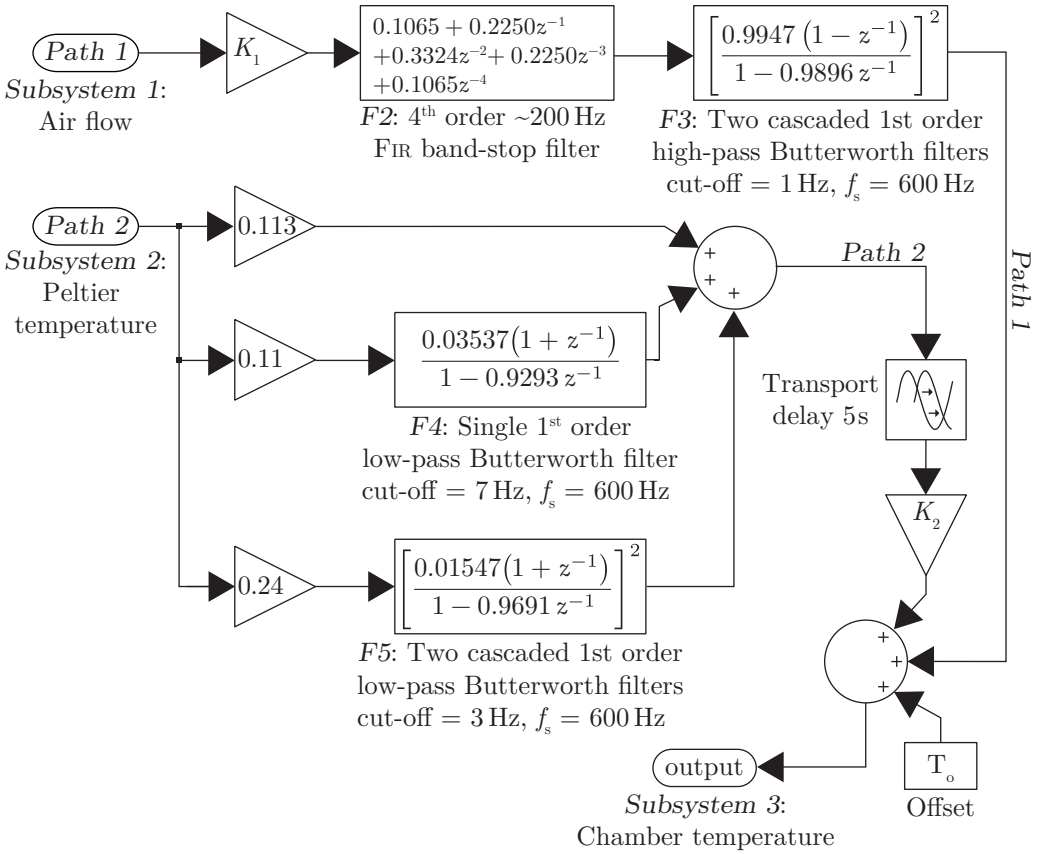


Figure 8.3: Overview of the Peltier cooling system

data output frequency spectrum in the *Benchmark*. For the *Subsystem 2* pathway, the signal was filtered by a first-order Butterworth low pass filter *F4* with cut-off frequency of 7/600, in parallel with a second order low pass filter *F5* cascaded by two first order Butterworth filters in series, with cut-off frequencies of 3/600. The signal itself was then added to its two filtered counterparts with different weightings, as shown in Fig. 8.3. A transport delay (pure time delay) was also incorporated in this pathway (see Section 8.3.2).

After combining the contributions from the two pathways, a constant offset T_o was added. This was based on comparing the simulated output with the mean value of the training data time domain response. It is reasonable to assume the offset temperature was related to the ambient temperature when the measurement took place (Cham, Tan & Tan, 2010).

8.2 Simulation

The simulation was performed using 3020 seconds (more than 6 periods) of data, with the inputs being the periodic multisine signals *A* & *B* scaled for *Subsystems 1* and *2* as specified in the *Benchmark*. To simulate the output for the training data, *Multisine A* was scaled with a range of 75° to 225° for the desired valve angle, while *Multisine B* was scaled with a range of 0 to 1 as the duty ratio. For the validation data, these signals were rescaled and swapped, with *Multisine B* being applied to *Subsystem 1* and *Multisine a* being applied to *Subsystem 2*. In each case, the output was taken from the sixth simulation period ($t = 2400$ to $2999\,\mathrm{s}$) to minimise transient effects. The outputs for both training data and validation data cases were compared with the respective averaged outputs from the experiments performed in the original Benchmark.

There is an overall positive system time delay of about 3 seconds observable in the simulation output of the master system, i.e. the simulation output lagged behind the true output. This indicates the presence of an overall negative delay in the model suggesting non-causality in signal *Path 1*. To compensate for this, the sample output was taken 3 samples later for the least error match.

8.3 Results and discussions

The models developed were used to simulate the system for both the training data and the validation data. While the structure of the models for *Subsystems 1* and *2* (Fig. 8.2) was the same for both sets of data, it was found to be necessary to make some small changes to the gains of the two pathways in the Subsystem 3 model (Fig. 8.3) between the two sets of data in order to achieve good results for both sets of data. For the training data, $K_1 = 0.09$ and $K_2 = 0.95$, while for the validation data $K_1 = 0.12$ and $K_2 = 0.99$. It was also necessary to make a small change to the constant temperature offset T_o between the two sets of data. The value was obtained by comparing the mean of the simulation output with that of the actual output and for the training data, $T_\mathrm{o} = 12.80\,^\circ\mathrm{C}$, whereas for the validation data, $T_\mathrm{o} = 12.73\,^\circ\mathrm{C}$.

8.3.1 *Path 1*: Signal path of *Subsystem 1* output in *Subsystem 3*

When simulating Subsystem 1, it was found that the contribution of *Path 1* was about 25% lower in the training data case than that in the validation data case (the respective values of K_1 being 0.09 and 0.12).

8.3.2 *Path 2*: Signal path of *Subsystem 2* output in *Subsystem 3*

Through simulating *Subsystem 2*, it was found that the contribution of *Path 2* in the validation data case is about 4% higher than that in the training data case (the respective values of K_2 being 0.99 and 0.95). In the physical realm, it is likely that the ambient temperature affects the gain due to effect of thermal gradient in heat dissipation of the physical system. More datasets including ambient temperature variations are required to verify this aspect of the model.

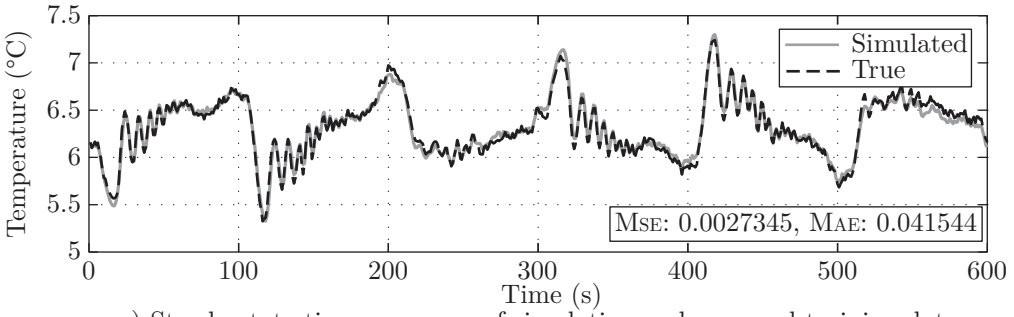
Another notable finding is that by observing the simulated output and the training data output set, it was found that *Path 2* has a time delay of about 5 seconds with respect to *Path 1*. This is independent of the 3 second delay mentioned in Section 8.2. Since the input and output signals are both sampled at 1 second intervals, a higher accuracy for the time delay cannot be attained. By physical reasoning, the time delay for heat dissipation will depend on ambient temperature as well as the immediate temperature output of *Subsystem 2*; so that it would not be surprising if the system is time varying. Nevertheless, for the similar input signal, it seems that the time delay of 5 seconds for *Path 2* is justified for both datasets.

8.3.3 Time responses and frequency response magnitudes

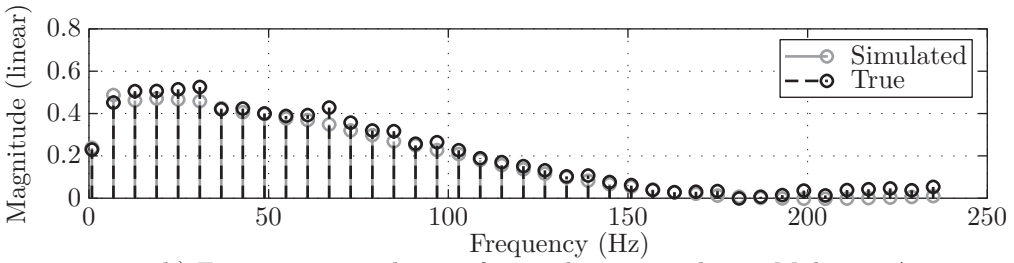
The training data results are shown in Fig. 8.4, with the corresponding results for the validation data in Fig. 8.5. In both figures, Panel (a) shows the time responses of the simulated system output and the actual averaged data, over one period of the two perturbation signals. Panels (b) and (c) show the frequency response magnitude contributions from *Subsystems 1* and *2* respectively.

It is possible to separate these contributions out for the actual data, because *Multisines A* and *B* have independent and thus separable frequency components, as outlined at the beginning of this chapter. Panel (d) shows the individual contributions to the time response from *Subsystems 1* and *2*.

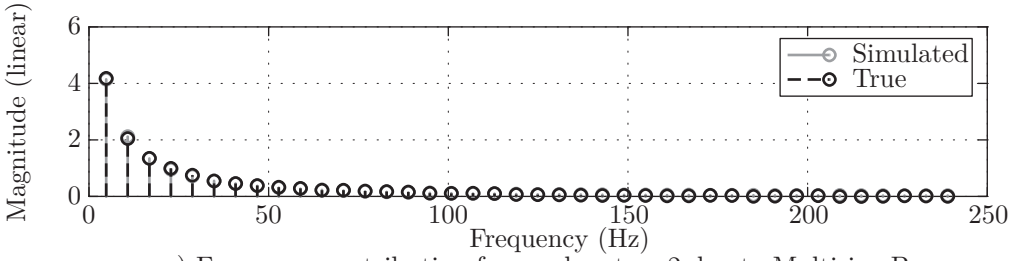
The mean squared error (MSE) and mean absolute error (MAE) for the training data were $2.73 \times 10^{-3}\,^{\circ}\mathrm{C}^2$ and $4.15 \times 10^{-2}\,^{\circ}\mathrm{C}$ respectively, while for the validation data they were $7.73 \times 10^{-3}\,^{\circ}\mathrm{C}^2$ and $7.52 \times 10^{-2}\,^{\circ}\mathrm{C}$. For the validation dataset, there appears to be some small time variation over the period of the data, and this may well be the reason for the somewhat higher error values with this set. From Fig. 8.5(a), the simulation slightly over-estimates the true output near the start of the period and slightly under-estimates it near the end of the period. The effect of time variation was not incorporated into the model.



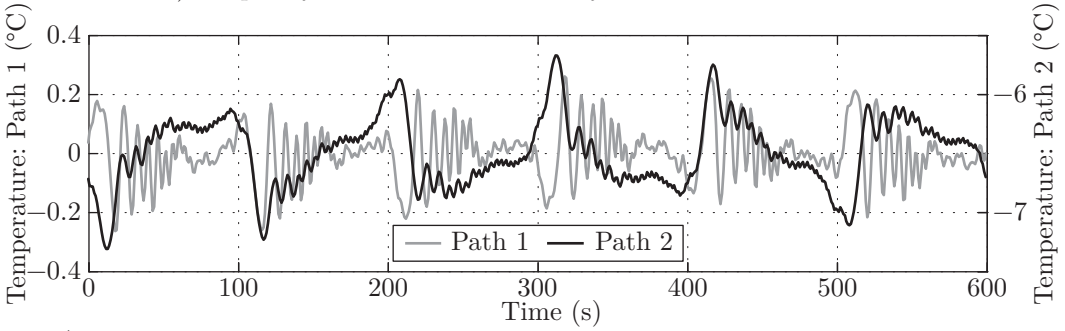
a) Steady state time responses of simulation and averaged training data



b) Frequency contribution from subsystem 1 due to Multisine A

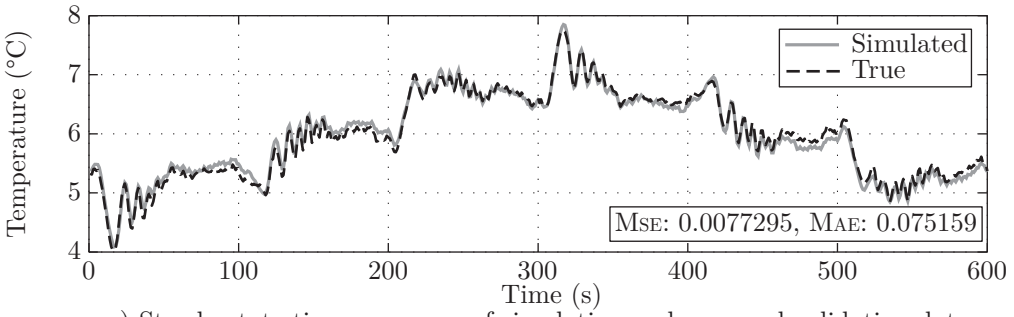


c) Frequency contribution from subsystem 2 due to Multisine B

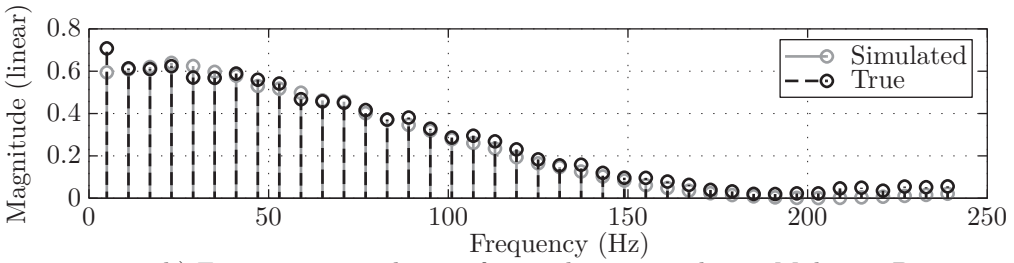


d) Individual contributions of time response disturbances from Subsystems 1 and 2

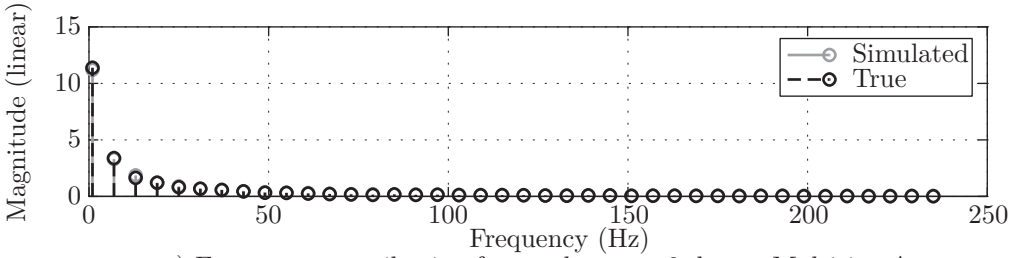
Figure 8.4: Results for the training dataset



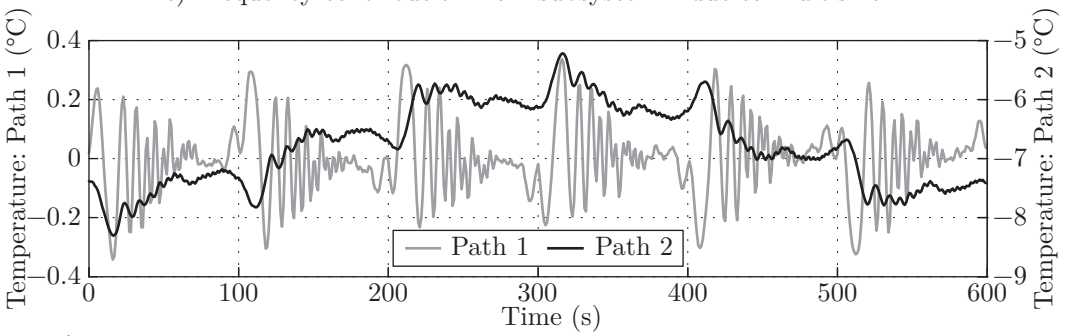
a) Steady state time responses of simulation and averaged validation data



b) Frequency contribution from subsystem 1 due to Multisine B



c) Frequency contribution from subsystem 2 due to Multisine A

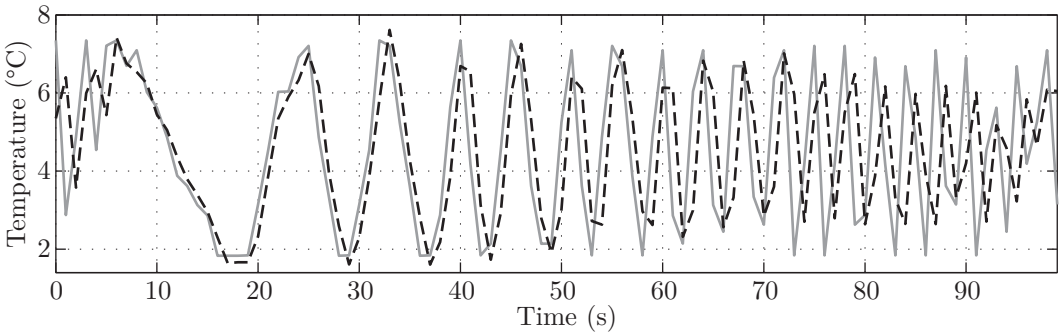


d) Individual contributions of time response disturbances from Subsystems 1 and 2

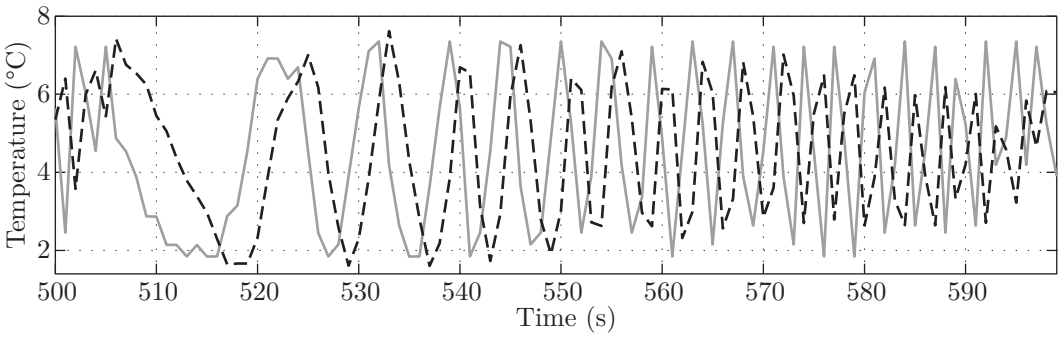
Figure 8.5: Results for the validation dataset

8.3.4 Intermediate signals

The validation dataset provided in Cham, Tan and Tan (2010) includes the intermediate output signals for *Subsystems 1* and *2*; these are intended for verification purposes only. Comparisons between these two signals predicted from the simulation and the actual signals given in the *Benchmark* are shown in Figures 8.6 and 8.7.



(a) First 100 samples ($t = 0$–$99\,\mathrm{s}$)



(b) Last 100 samples ($t = 500$–$599\,\mathrm{s}$)

Figure 8.6: Intermediate signal for *Subsystem 1* for simulation and experiment data, first and last 100 samples
Grey line: Simulation, dashed solid line: Measurement of the true system

For clarity, with respect to the sampling interval of one second, only the first 100 and the last 100 samples of the output for *Subsystem 1* are shown in Fig. 8.6. This is sufficient to illustrate the variability in time delay between the simulation and the experiment data. Near $t = 0\,\mathrm{s}$ the delay is negligible, but it then gradually increases, reaching about 1 sampling interval near $t = 99\,\mathrm{s}$, and about 2 sampling intervals by the end of the period ($t = 599\,\mathrm{s}$). This is not accounted for in the current model. It can be seen in Fig. 8.7 that the model of *Subsystem 2* gives a remarkably good fit throughout the full period of the data. From Fig. 8.5(d), it can be seen

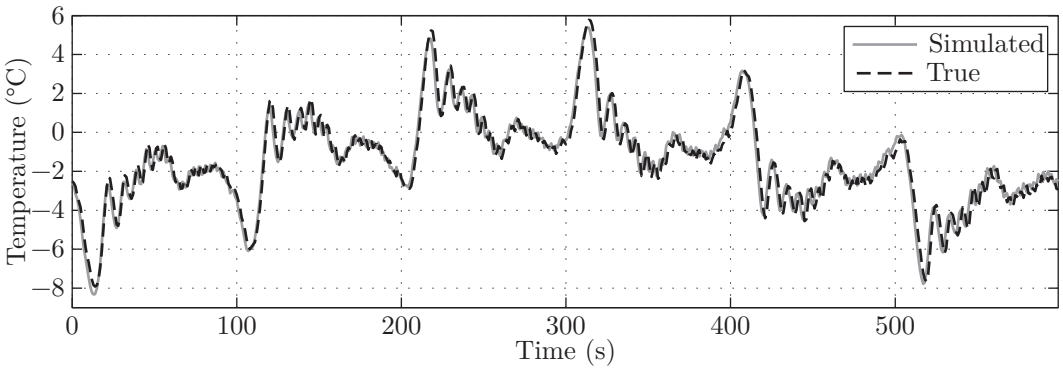


Figure 8.7: Intermediate signal for *Subsystem 2* for simulation and experiment data

that the contribution of *Subsystem 1* to the overall system output is considerably smaller than that of *Subsystem 2*. This means that the model developed provided a good fit to the overall system output for both the training data and the validation data, even though the modelling of *Subsystem 1* was not particularly accurate.

8.4 Comparison with other approaches

Results from other approaches on the same benchmark exercise are given in Table 8.1. It can be seen that the current 'hybrid empirical and mechanistic' approach—a mix between blackbox and whitebox modelling, has resulted in good performance while retaining the physical interpretations of the system structure.

Table 8.1: Comparison between several models

Source	Model	MAE ($\times 10^{-3}$)	MSE ($\times 10^{-3}$)
Cham, Tan and Ramar (2010)	Variable time delay	59.1	5.71
Larkowski and Burnham (2010)	Box-Jenkins	56.5	5.05
Marconato et al. (2010)	Parametric Best Linear Approximation (BLA)	73.2	8.40
Taylor and Young (2010)	Data-based mechanistic	81	10
Wong and Godfrey (2010)	Hybrid empirical and mechanistic	75.2	7.73

8.5 Conclusions

This chapter is concerned with the the 'hyperfast' switching Peltier cooling system benchmark, first proposed in the U.K. Automatic Control Council (UKACC) International Conference on Control, 2010 (Control 2010). The chapter has illustrated how a simple modelling exercise using MATLAB *Simulink* can provide valuable insights into the structure of a system such as the time delay of a signal path, when some physical information is known beforehand. The models for *Subsystems 2* and *3* were linear. The principal nonlinearities in the actual system were the stepper motor and the steady-state gain characteristic between angle and flow rate in *Subsystem 1* as stated in Cham, Tan and Tan (2010), which could be modelled readily in *Simulink*, and this meant that the other parts of modelling could be reduced to a simple linear fitting exercise.

It was necessary to make small changes to the gains K_1 and K_2 of the paths from *Subsystems 1* and *2* and to the temperature offset T_o in the model between the training data and the validation data. This is not unreasonable, since the two sets of data were collected at different times, and in all probability, the ambient conditions were not quite the same between the measurements.

Chapter 9

Conclusions and Future Work

All systems are nonlinear to some extent, and while nonlinear modelling techniques exist, it is sometimes preferable to apply linearisation techniques as linear models, especially when applied to control theory, are well understood. The Best Linear Approximation (Bla) is one linearisation technique that achieves the least total sum squared error when approximating a time-invariant nonlinear system.

It is known that the Best Linear Approximation (Bla) is dependent upon the power spectrum and the amplitude distribution of the input. The Bla obtained from input signals or sequences having a Gaussian amplitude distribution, called the Gaussian Bla, has certain desirable properties, some of which are well known. For example, in the frequency domain, the Bla is composed of the combined linear dynamics of all the linearities in a block-structured system, so that it is possible to separate them in terms of their poles and zeros (see Chapter 7 for a modelling exercise utilising this property). However, this is not the case for a Bla estimated by non-Gaussian inputs.

The dependence on the amplitude distribution has not previously been studied in detail in the literature. In Chapter 3 of this thesis, the theoretical closed-form expressions were derived for the Blas for both discrete-time Wiener-Hammerstein (Wh) systems and discrete-time Volterra systems for inputs with an arbitrary amplitude distribution. The theory is valid for binary and ternary sequences for example, but they must have white power spectra and zero mean. In practice, the zero-mean requirement is not of concern, as having non-zero mean only introduces a constant bias in the operating point about which the Bla is linearised, and does not affect the estimated dynamics. The developed theory has been verified by both simulation experiments in Chapter 3 as well as a set of physical experiments performed on an

electronic Wiener system in Chapter 6. The theory has shown that the Gaussian and non-Gaussian BLA indeed differ, and the discrepancy is due to the higher order moment terms of the input. A relative measure to quantify the discrepancy, called the Discrepancy Factor (DF), was proposed in Section 3.5. It was found that in general, the DF decreases with the memory of the linear dynamics of the nonlinear system. In addition, a method was proposed to design discrete multilevel sequences for Gaussianity, in order to minimise the DF. This was shown in Chapter 5 to be viable and effective; more so than the commonly used uniformly distributed sequences. This has obvious implications in experiment design if the objective is to extract the Gaussian BLA, especially when the principal degrees of the nonlinearities are known.

Lastly, Chapters 7 and 8 document benchmark studies which provided valuable training opportunities for the author in system identification and as general contributions to the academic discussion in the conferences where the work was presented.

9.1 Research impact

There are systems where applying continuous level signals, for example, Gaussian inputs, is either impractical or impossible. For instance, if the actuator of a system is a valve that can only be opened or closed, only a binary input may be applied. The thesis has provided theoretical insights to the identification of nonlinear systems with respect to the amplitude distribution of the input. The discrete-time theory of the BLA developed in Chapter 3 is applicable to a large variety of systems, thanks to the generalisation of the closed-form expressions to the Volterra theory. In addition, the amplitude distribution of the inputs is arbitrary, allowing the theoretical BLA to be studied for any given input. This piece of theoretical work sheds light on the behaviour of the BLA when the amplitude distribution (or the higher order moments) of the input are non-Gaussian. This, coupled with the method of designing multilevel sequences for Gaussianity in Chapter 5, allow the Gaussian BLA to be estimated more accurately with discrete-level signals. The research can benefit potential applications where discrete or digital input signals predominate, for instance, identification of digital communication channels, characterisation of delta-sigma digital-to-analogue (DAC) converters, chemical processes involving valve control or heating elements, and micro-fluidics in micro-electromechanical (MEM) systems.

When a Maximum Length Binary Sequence (MLBS) is used to identify the BLA of a nonlinear system, Chapter 4 has demonstrated that the nonlinear distortions appear structured, which can be taken advantage of—instead of using traditional mean averaging, the use of median averaging can potentially offer great benefits in terms of the accuracy of the obtained model. There is a risk of obtaining worse estimates when the signal-to-noise ratio (SNR) is low, but the potential gain has been shown to outweigh the loss by a large margin, even in a scenario where the long system memory favours the mean estimator. Since the median estimator is simple to apply, this is worth considering and should be exploited in many applications.

9.2 Future work

1. Extend the BLA theory developed in Chapter 3 for non-white inputs.
   - ► One of the assumptions made in Chapter 3 was that the input used to estimate the BLA is assumed white, so that its auto-correlation is a Kronecker delta function. This requirement can be relaxed somewhat if a non-white input can be assumed to be generated by filtering a white input through a causal and stable filter. Unlike Gaussian inputs where dynamics are simply cascaded, the non-Gaussian BLA will be affected by an input filter that introduces extra dynamics. The non-Gaussian BLA obtained from a coloured input will be different to that obtained from a white input, even with an identical amplitude distribution. Investigating this aspect is one possibility for future work.
2. Extend the discrete-time BLA theory developed in Chapter 3 to the continuous-time domain.
   - ► In light of the difficulty encountered with continuous-time nonlinear effects of the input actuator observed with the physical experiment in Chapter 6, some preliminary work was performed in an attempt to extend the BLA theory to the continuous-time domain. The approximated closed form solutions of the BLA theory in the continuous-time domain suggests far more complex expressions instead of the relatively simple results seen for the discrete-time counterparts in Chapter 3, for even 3^{rd} degree nonlinearities.

3. Investigate the merits of other types of location estimators, for instance, weighted median, for averaging out the structured nonlinear distortions when Maximum Length Binary Sequences (MLBS's) are used (Section 4.5).
4. Formulate a statistical study of the trade-off between the Discrepancy Factor (DF) (Section 3.5) and the variance due to the nonlinear distortions.
   - When estimating the Gaussian BLA of a nonlinear system with MLBS's, the discrepancy between the binary and Gaussian BLAs can be treated as a bias term that is dependent on time or frequency. On the other hand, the structured nonlinear distortions change with different realisations and this variation introduces a variance. It has been shown in Section 3.5.2 that the DF is dependent upon the memory of the system relative to the sampling time; the shorter it is, the higher the discrepancy. It is also found that, generally speaking, systems with a longer memory make nonlinear distortion less structured (Section 4.5) and more similar to background noise, which decrease the effectiveness of the median estimator. Hence, the bias and variance aspect is a trade-off. It is perhaps beneficial to perform a statistical analysis to formulate a guideline on the recommended experiment design with respect to the sampling time, in order to have a low DF and effective rejection of the nonlinear distortion with the median estimator when MLBS's are used.
5. Evaluate the use of the robust BLA estimation procedure of Section 4.2 in the context of Section 4.5—to apply mean averaging over P and median averaging over M so that the optimality of mean averaging for driving out Gaussian environment noise and the effectiveness of median in averaging out structured nonlinear distortions can both be exploited.
6. Consider extending the theory to multiple-input multiple-output (MIMO) systems.
7. Perform further experiment verification of the BLA theory on practical problems.
8. Investigate whether the use of higher order cumulants instead of higher order moments can simplify the algebra for the discrepancy between the Gaussian and non-Gaussian BLA—the higher than third order cumulants for a Gaussian amplitude distribution are uniquely zero, unlike any other distributions.

9.3 Summary of contributions

This thesis provides the following major contributions:

1. A general theory has been developed to study the discrepancy between the Gaussian and non-Gaussian Best Linear Approximation (BLA) with respect to the amplitude distribution of the non-Gaussian input. It has been shown that the BLA directly depends on the higher order moments of the excitation signal.

2. Discrepancy Factor (DF), the measure used to quantify the difference between a Gaussian BLA and non-Gaussian BLA has been developed; in general the DF depends on the dynamics of the system—the longer the system memory, the lower the DF.

3. The theory of the BLA has been successfully verified by both simulation experiments, and a physical experiment performed on an electronic circuit built to emulate a discrete-time Wiener system.

4. A method has been proposed to capitalise on the structured nonlinear distortions that appear when estimating the BLA of a system using, specifically, MLBS's as input excitations. By simply switching the averaging scheme from traditional mean averaging to median averaging in the time domain, it has been shown that the nonlinear distortions can be subdued effectively, depending on the signal-to-noise ratio (SNR).

5. A method for designing random multilevel sequences for Gaussianity has been proposed to decrease the DF. By suitably matching the Gaussian moments, it has been demonstrated that the designed sequences are effective in subduing the DF; more so than the commonly used uniformly distributed signals.

References

Amrani, M. E. H., Dowdeswell, R. M., Payne, P. A. & Persaud, K. (1998, April). Pseudo-random binary sequence interrogation technique for gas sensors. *Sensors and Actuators B: Chemical*, *47*(1–3), 118–124. (Cit. on p. 19).

Analog Devices. (2008). AD8610: Precision, very low noise, low input bias current, wide bandwidth JFET operational amplifier. [Datasheet]. Norwood, MA. (Cit. on p. 103).

Barker, H. A. (1967). Choice of pseudorandom binary signals for system identification. *Electronics Letters*, *3*, 524–526. (Cit. on p. 18).

Barker, H. A. (1993). Design of multi-level pseudo-random signals for system identification. In K. R. Godfrey (Ed.), *Perturbation signals for system identification* (Chap. 11, pp. 325–326). Hemel Hempstead, Herts, U.K.: Prentice Hall. (Cit. on p. 163).

Barker, H. A. (2008, August). GALOIS - a program to generate pseudorandom signals. Retrieved from http://www.eng.warwick.ac.uk/eed/dsm/galois/. (Cit. on pp. xxii, 18, 24)

Barker, H. A. & Godfrey, K. R. (1999, June). System identification with multi-level periodic perturbation signals. *Control Engineering Practice*, *7*(6), 717–726. (Cit. on p. 85).

Barker, H. A., Tan, A. H. & Godfrey, K. R. (2009, July). Object-oriented input signal design for system identification. In *Proc. 15th IFAC Symposium on System Identification*, 6th–8th July 2009 (pp. 180–185). St. Malo, France. (Cit. on p. 95).

Bendat, J. S. & Piersol, A. G. (1980, May). *Engineering applications of correlation and spectral analysis*. New York: Wiley. (Cit. on pp. 29, 58).

Blackman, R. B. & Tukey, J. W. (2012, July). *The measurement of power spectra, from the point of view of communications engineering*. Whitefish, MT: Literary Licensing, LLC. (Cit. on p. 13).

Borish, J. & Angell, J. B. (1983). An efficient algorithm for measuring the impulse response using pseudorandom noise. *J. Audio Eng. Soc. 31*(7/8), 478–488. (Cit. on p. 19).

Boyd, S. (1986, October). Multitone signals with low crest factor. *IEEE Trans. Circuits Syst. 33*(10), 1018–1022. (Cit. on p. 17).

Bussgang, J. J. (1952, March). *Cross-correlation function of amplitude-distorted Gaussian signals* (tech. rep. No. 216). Res. Lab. Elec., Mas. Inst. Technol. Cambridge, MA. (Cit. on pp. 5, 31).

Cham, C. L., Tan, A. H. & Ramar, K. (2010, September). Modelling of a hyper-fast switching Peltier cooling system using a variable time delay model. In *Proc. UKACC International Conference on Control*, 7th–10th September 2010 (pp. 179–184). Coventry, U.K. (Cit. on p. 145).

Cham, C. L., Tan, A. H. & Tan, W. H. (2010, September). Hyperfast switching Peltier cooling system benchmark. In *Proc. UKACC International Conference on Control*, 7th–10th September 2010 (pp. 185–190). Coventry, U.K. (Cit. on pp. 8, 135–139, 144, 146).

Coleman, T., Branch, M. & Grace, A. (1999). *Optimization Toolbox for use with MATLAB*. The MathWorks Inc. Natick, MA. (Cit. on p. 116).

Dobrowiecki, T. P. & Schoukens, J. (2002, May). Cascading Wiener-Hammerstein systems. In *Proc. IEEE Instrumentation and Measurement Technology Conference*, 21st–23rd May 2002 (pp. 881–886). Anchorage, AK. (Cit. on pp. 33, 116).

Dodge, Y., Cox, D., Commenges, D., Davison, A., Solomon, P. & Wilson, S. (Eds.). (2006, September 7). *Central tendency*. New York, NY: Oxford University Press, (cit. on p. 74).

Enqvist, M. (2005). *Linear models of nonlinear systems* (Doctoral dissertation, Linköping University, Sweden). (Cit. on pp. 5, 28, 29, 91).

Enqvist, M. & Ljung, L. (2005, December). *Linear approximations of nonlinear FIR systems for separable input processes* (tech. rep. No. LiTH-ISY-R-2718). Department of Electrical Engineering, Linköping University. Linköping, Sweden. (Cit. on pp. 27–29).

Eykhoff, P. (1974, May). *System identification—parameter and state estimation*. London, U.K.: Wiley-Interscience. (Cit. on p. 29).

Falck, T., Dreesen, P., De Brabanter, K., Pelckmans, K., De Moor, B. & Suykens, J. A. K. (2012, November). Least-squares Support Vector Machines for the identification of Wiener-Hammerstein systems. *Control Engineering Practice*, *20*(11), 1165–1174. (Cit. on p. 133).

Fallah-Jamshidi, S., Amiri, M. & Karimi, N. (2010, September). Nonlinear continuous multi-response problems: a novel two-phase hybrid genetic based metaheuristic. *Applied Soft Computing*, *10*(4), 1274–1283. (Cit. on p. 121).

Fréchet, M. R. (1910). Sur les fonctionnelles continues. *Annales Scientifiques de l'École Normale Supérieure*. 3rd ser., *27*, 193–216. (Cit. on p. 43).

Frigyik, B. A., Srivastava, S. & Gupta, M. R. (2008, November). Functional Bregman divergences and Bayesian estimation of distributions. *IEEE Trans. on Inf. Theory*, *54*(11), 5130–5139. (Cit. on p. 86).

Godfrey, K. R. (1980, September). Correlation methods. *Automatica*, *16*(5), 527–534. (Cit. on pp. 16, 72).

Godfrey, K. R. & Moore, D. J. (1974). Identification of processes having direction-dependent responses, with gas-turbine engine applications. *Automatica*, *10*(5), 469–481. (Cit. on pp. 9, 19, 69, 74).

Godfrey, K. R. (1993). Introduction to perturbation signals for time-domain system identification. In K. R. Godfrey (Ed.), *Perturbation signals for system identification* (Chap. 1). Hemel: Prentice-Hall. (Cit. on pp. 18, 19).

Godfrey, K. R., Barker, H. A. & Tucker, A. J. (1999, November). Comparison of perturbation signals for linear system identification in the frequency domain. *IEE Proc. Control Theory and Applications*, *146*(6), 535–548. (Cit. on pp. 14, 95).

Godfrey, K. R., Tan, A. H., Barker, H. A. & Chong, B. (2005, November). A survey of readily accessible perturbation signals for system identification in the frequency domain. *Control Engineering Practice*, *13*(11), 1391–1402. (Cit. on pp. xxii, 18, 24, 85, 181).

Golomb, S. W. (1981, June). *Shift register sequences* (Revised edition). Laguna Hills, CA: Aegean Park Press. (Cit. on pp. 18, 19).

Guillaume, P., Schoukens, J., Pintelon, R. & Kollár, I. (1991, December). Crest-factor minimization using nonlinear chebyshev approximation methods. *IEEE Trans. Instrum. Meas. 40*(6), 982–989. (Cit. on p. 18).

Hartmann, W. M. (1997, January 9). *Signals, sound, and sensation*. New York, NY: Springer. (Cit. on p. 23).

Hirotugu, A. (1974, December). A new look at the statistical model identification. *IEEE Trans. Auto. Cont. 19*(6), 716–723. (Cit. on p. 114).

IEEE Acoustics, Speech and Signal Processing Society. Digital Signal Processing Committee. (1979). *Programs for digital signal processing*. New York: IEEE Press. (Cit. on p. 65).

Isserlis, L. (1916, May). On certain probable errors and correlation coefficients of multiple frequency distributions with skew regression. *Biometrika*, *11*(3), 185–190. (Cit. on p. 169).

Isserlis, L. (1918, November). On a formula for the product-moment coefficient of any order of a normal frequency distribution in any number of variables. *Biometrika*, *12*(1), 134–139. (Cit. on p. 169).

Kenney, J. F. & Keeping, E. S. (1962). The median. In *Mathematics of statistics* (3rd ed., pp. 211–212). Princeton, NJ: Van Nostrand. (Cit. on pp. 74–76).

Klock, H. & Buhmann, J. M. (2000, April). Data visualization by multidimensional scaling: a deterministic annealing approach. *Pattern Recognition*, *33*(4), 651–669. (Cit. on p. 121).

Kollár, I. (1994). *Frequency Domain System Identification Toolbox for use with MATLAB*. MathWorks, Inc. Natick, MA. (Cit. on pp. xxii, 91, 104, 114).

Korenberg, M. J. (1991, July). Parallel cascade identification and kernel estimation for nonlinear systems. *Ann. Biomed. Eng. 19*(4), 429–455. (Cit. on pp. 36, 43).

Krauss, T. P., Shure, L. & Little, J. (1994). *Signal Processing Toolbox for use with MATLAB*. MathWorks, Inc. Natick, MA. (Cit. on pp. xxiv, 138).

Kullback, S. (1987, November). Letter to the editor: the Kullback-Leibler distance. *The American Statistician*, *41*(4), 340–341. (Cit. on p. 86).

Larkowski, T. & Burnham, K. J. (2010, September). Identification of a discrete-time parametric model of a hyperfast switching Peltier cooling system. In *Proc. UKACC International Conference on Control*, 7th–10th September 2010 (pp. 607–612). Coventry, U.K. (Cit. on p. 145).

Ljung, L. (1999, January). *System identification: theory for the user* (2nd). Hemel: Prentice-Hall. (Cit. on p. 3).

Lopes dos Santos, P., Ramos, J. A. & Martins de Carvalho, J. L. (2012, November). Identification of a benchmark Wiener-Hammerstein: A bilinear and Hammerstein-bilinear model approach. *Control Engineering Practice*, *20*(11), 1156–1164. (Cit. on p. 133).

Mäkilä, P. M. (2004, July). On optimal LTI approximation of nonlinear systems. *IEEE Trans. Auto. Cont. 49*(7), 1178–1182. (Cit. on p. 27).

Mäkilä, P. M. (2006, June). LTI approximation of nonlinear systems via signal distribution theory. *Automatica*, *42*, 917–928. (Cit. on p. 27).

Mäkilä, P. M. & Partington, J. R. (2004, July). Least-squares LTI approximation of nonlinear systems and quasistationarity analysis. *Automatica*, *40*, 1157–1169. (Cit. on p. 27).

Marconato, A., Van Mulders, A., Pintelon, R., Rolain, Y. & Schoukens, J. (2010, September). A simple and fast technique to evaluate the possibility of approximating a nonlinear system by a nonlinear PISPO model. In *Proc. UKACC International Conference on Control*, 7th–10th September 2010 (pp. 674–679). Coventry, U.K. (Cit. on pp. 11, 145).

Marmarelis, V. Z. (2004). *Nonlinear dynamic modeling of physiological systems*. Hoboken, NJ: Wiley-IEEE Press. (Cit. on p. 19).

Michalowicz, J. V., Nichols, J. M., Bucholtz, F. & Olson, C. C. (2009, July). An isserlis' theorem for mixed gaussian variables: application to the auto-bispectral density. *Journal of Statistical Physics*, *136*(1), 89–102. (Cit. on p. 170).

Mitra, A. (2008). On the properties of pseudo noise sequences with a simple proposal of randomness test. *International Journal of Electrical and Computer Engineering*, *3*(3), 164–169. (Cit. on p. 23).

New Wave Instruments. (2010, April 6). Linear feedback shift registers—implementation, m-sequence properties, feedback tables. Retrieved January 13, 2013, from http://www.newwaveinstruments.com/resources/articles/m_sequence_linear_feedback_shift_register_lfsr.htm. (Cit. on p. 21)

Ng, Y. H., Tan, A. H. & Chuah, T. C. (2011, September). Channel identification of concatenated fiber-wireless uplink using ternary signals. *IEEE Trans. Vehicular Technology*, *60*(7), 3207–3217. (Cit. on p. 85).

Norton, J. P. (2009, April). *An introduction to identification* (Reprint). Mineola, New York, NY: Dover Publications. (Original work published 1986). (Cit. on p. 19)

Paduart, J., Lauwers, L., Pintelon, R. & Schoukens, J. (2012, November). Identification of a Wiener-Hammerstein system using the polynomial nonlinear state space approach. *Control Engineering Practice*, *20*(11), 1133–1139. (Cit. on p. 133).

Piegat, A. (2001). *Fuzzy modelling and control*. New York: Physica-Verl. (Cit. on p. 116).

Pinsky, M. A. (2002). *Introduction to fourier analysis and wavelets*. Pacific Grove, CA: Brooks/Cole. (Cit. on p. 13).

Pintelon, R. & Schoukens, J. (2012, April). *System identification: a frequency domain approach* (2nd ed.). Hoboken, NJ: Wiley-IEEE Press. (Cit. on pp. 11, 14, 27, 28, 33, 34, 53, 58–60, 63, 65, 68, 82, 86, 95, 104, 111, 116, 170).

Piroddi, L., Farina, M. & Lovera, M. (2012, November). Black box model identification of nonlinear input-output models: A Wiener-Hammerstein benchmark. *Control Engineering Practice*, *20*(11), 1109–1118. (Cit. on p. 133).

Rake, H. (1980, September). Step response and frequency response methods. *Automatica*, *16*(5), 519–526. (Cit. on p. 58).

Reed, M. J. & Hawksford, M. O. J. (1996, October). Identification of discrete Volterra series using maximum length sequences. *IEE Proceedings Circuits, Devices & Systems*, *143*(5), 241–248. (Cit. on p. 19).

Rosenqvist, F., Tan, A. H., Godfrey, K. R. & Karlström, A. (2006). Direction-dependent system modelling approaches exemplified through an electronic nose system. *IEEE Trans. Control Syst. Technol. 14*(3), 526–531. (Cit. on p. 69).

Schetzen, M. (2006, April). *The Volterra and Wiener theories of nonlinear systems* (Revised). Malabar, FL: Krieger Publishing Company. (Original work published 1980). (Cit. on pp. 31, 169)

Schoukens, J., Dobrowiecki, T., Rolain, Y. & Pintelon, R. (2009, May). Upper bounding the variations of the best linear approximation of a nonlinear system in a power sweep measurement. In *Proc. IEEE International Instrumentation and Measurement Technology Conference 2009*, 5th–7th May 2009 (pp. 1514–1519). Singapore. (Cit. on p. 18).

Schoukens, J., Lataire, J., Pintelon, R., Vandersteen, G. & Dobrowiecki, T. (2009, May). Robustness issues of the best linear approximation of a nonlinear system. *IEEE Trans. Instrum. Meas. 58*(5), 1737–1745. (Cit. on pp. 18, 28, 33, 34, 86, 111, 116).

Schoukens, J., Pintelon, R., Dobrowiecki, T. & Rolain, Y. (2005, March). Identification of linear systems with nonlinear distortions. *Automatica*, *41*(3), 491–504. (Cit. on p. 28).

Schoukens, J., Pintelon, R. & Renneboog, J. (1988, March). A maximum likelihood estimator for linear and nonlinear systems—a practical application of estimation techniques in measurement problems. *IEEE Trans. Instrum. Meas. 37*, 10–17. (Cit. on p. 113).

Schoukens, J., Pintelon, R. & Rolain, Y. (2012). *Mastering system identification in 100 excercises*. Hoboken, NJ: Wiley-IEEE Press. (Cit. on pp. 17, 182).

Schoukens, J., Pintelon, R., van der Ouderaa, E. & Renneboog, J. (1988, September). Survey of excitation signals for FFT based signal analyzers. *IEEE Trans. Instrum. Meas. 37*(3), 342–352. (Cit. on pp. 14, 18, 95).

Schoukens, J., Suykens, J. A. K. & Ljung, L. (2009, July). Wiener-Hammerstein benchmark. In *Proc. 15th IFAC Symposium on System Identification*, 6th–

8th July 2009. St. Malo, France. Retrieved from http://tc.ifac-control.org/1/1/Data%20Repository/sysid-2009-wiener-hammerstein-benchmark. (Cit. on pp. xviii, 7, 111, 129, 132)

Schoukens, J., Swevers, J., Pintelon, R. & van der Auweraer, H. (2004, July). Excitation design for FRF measurements in the presence of non-linear distortions. *Mechanical Systems and Signal Processing*, *18*(4), 727–738. (Cit. on p. 24).

Schroeder, M. (1970, January). Synthesis of low-peak-factor signals and binary sequences with low autocorrelation (corresp.) *IEEE Trans. Inf. Theory*, *16*(1), 85–89. (Cit. on pp. 17, 18).

Serfling, R. (2011). Asymptotic relative efficiency in estimation. In M. Lovric (Ed.), *International encyclopedia of statistical science* (Vol. 1, pp. 68–72). London: Springer. (Cit. on pp. 75–77).

Shimkin, N. & Feuer, A. (1987, May). Persistency of excitation in continuous-time systems. *Systems & Control Letters*, *9*(3), 225–233. (Cit. on pp. 14, 34).

Simon, M. K., Omura, J. K., Scholtz, R. A. & Levitt, B. K. (1994). *Spread spectrum communications handbook*. New York: McGraw-Hill. (Cit. on p. 19).

Sjöberg, J., Lauwers, L. & Schoukens, J. (2012, November). Identification of Wiener-Hammerstein models: Two algorithms based on the best split of a linear model applied to the SYSID'09 benchmark problem. *Control Engineering Practice*, *20*(11), 1119–1125. (Cit. on p. 133).

Söderström, T. & Stoica, P. (1989). *System identification*. Hemel Hempstead, Herts, U.K.: Prentice Hall. (Cit. on pp. 14, 34).

Stone, M. H. (1948). The generalized Weierstrass approximation theorem. *Mathematics Magazine*, *21*(4), 167–184. (Cit. on p. 43).

Sumarac-Pavlovic, D., Mijic, M. & Kurtovic, H. (2008, April). A simple impulse sound source for measurements in room acoustics. *Applied Acoustics*, *69*(4), 378–383. (Cit. on p. 4).

Sutter, E. E. (1987). A practical non-stochastic approach to nonlinear time-domain analysis. In V. Z. Marmarelis (Ed.), *Advanced methods of physiological system modeling* (Vol. 1, pp. 303–315). Los Angeles, CA: Biomedical Simulations Resource, University of Southern California. (Cit. on p. 19).

Sutter, E. E. (1992). A deterministic approach to nonlinear systems analysis. In R. B. Pinter & B. Nabet (Eds.), *Nonlinear vision: determination of neural receptive fields, function, and networks* (pp. 171–220). Boca Raton, FL: CRC Press. (Cit. on p. 19).

Tan, A. H. & Godfrey, K. R. (2002, August). The generation of binary and near-binary pseudorandom signals: an overview. *IEEE Trans. Instrum. Meas. 51*, 583–588. (Cit. on pp. xxiv, 24, 181).

Tan, A. H. & Godfrey, K. R. (2004). Modeling of direction-dependent processes using Wiener models and neural networks with nonlinear output error structure. *IEEE Trans. Instrum. Meas. 53*(3), 744–753. (Cit. on p. 85).

Tan, A. H., Wong, H. K. & Godfrey, K. R. (2012, November). Identification of a Wiener-Hammerstein system using an incremental nonlinear optimisation technique. *Control Engineering Practice*, *20*(11), 1140–1148. (Cit. on pp. 8, 111, 133).

Taylor, C. J. & Young, P. C. (2010, September). Captain Toolbox analysis of the hyperfast switching Peltier cooling system benchmark. In *Proc. UKACC International Conference on Control*, 7th–10th September 2010 (pp. 1082–1087). Coventry, U.K. (Cit. on p. 145).

Truong, N.-V. & Wang, L. (2009, July). Benchmark nonlinear system identification using wavelet based sdp models. In *Proc. 15th IFAC Symposium on System Identification*, 6th–8th July 2009 (pp. 1104–1109). St. Malo, France. (Cit. on p. 133).

van der Ouderaa, E., Schoukens, J. & Renneboog, J. (1988, March). Peak factor minimization, using a time-frequency domain swapping algorithm. *IEEE Trans. Instrum. Meas. 37*(1), 144–147. (Cit. on p. 18).

Vanderkooy, J. (1994, April). Aspects of mls measuring systems. *J. Audio Eng. Soc. 42*(4), 219–231. (Cit. on p. 9).

Venna, J. & Kaski, S. (2006, July). Local multidimensional scaling. *Neural Networks*, *19*(6), 889–899. (Cit. on p. 121).

Vilkko, M. & Roinila, T. (2008, June). Designing maximum length sequence signal for frequency response measurement of switched mode converters. In *Proc. Nordic Workshop on Power and Industrial Electronics*, 9th–11th June 2008. Helsinki Uni. of Technology. Espoo, Finland. (Cit. on p. 19).

Volterra, V. (2005, January). *Theory of functionals and of integral and integro-differential equations* (Reprint). Mineola, New York, NY: Dover Publications. (Original work published 1930). (Cit. on pp. 4, 43)

Westwick, D. T. & Kearney, R. E. (2003, August). Models of nonlinear systems. In *Identification of nonlinear physiological systems* (Chap. 4). Hoboken, NJ: Wiley-IEEE Press. (Cit. on p. 43).

Wills, A. & Ninness, B. (2012, November). Generalised Hammerstein-Wiener system estimation and a benchmark application. *Control Engineering Practice*, *20*(11), 1097–1108. (Cit. on pp. 130, 133).

Wong, H. K. & Godfrey, K. R. (2010, September). A greybox approach to modelling a hyperfast switching Peltier cooling system. In *Proc. UKACC International Conference on Control*, 7th–10th September 2010 (pp. 1200–1205). Coventry, U.K. (Cit. on pp. 8, 145).

Wong, H. K., Schoukens, J. & Godfrey, K. R. (2012a, March). Analysis of best linear approximation of a Wiener-Hammerstein system for arbitrary amplitude distributions. *IEEE Trans. Instrum. Meas. 61*(3), 645–654. (Cit. on pp. 8, 38, 48).

Wong, H. K., Schoukens, J. & Godfrey, K. R. (2012b, September). Experimental verification of best linear approximation of a Wiener system for binary excitations. In *Proc. UKACC International Conference on Control*, 3rd–5th September 2012 (pp. 941–946). Cardiff, U.K. (Cit. on p. 9).

Wong, H. K., Schoukens, J. & Godfrey, K. R. (2012c, July). The use of binary sequences in determining the best linear approximation of nonlinear systems. In *Proc. 16th IFAC Symposium on System Identification*, 11th–13th July 2012 (pp. 1323–1328). Brussels, Belgium. (Cit. on pp. 9, 63).

Wong, H. K., Schoukens, J. & Godfrey, K. R. (2013a, February). Design of multilevel signals for identifying the best linear approximation of nonlinear systems. *IEEE Trans. Instrum. Meas. 62*(2), 519–524. (Cit. on p. 9).

Wong, H. K., Schoukens, J. & Godfrey, K. R. (2013b). Structured non-linear noise behaviour and the use of median averaging in non-linear systems with m-sequence inputs. *IET Control Theory and Applications*. [Accepted for Publication]. doi:10.1049/iet-cta.2012.0622. (Cit. on pp. 9, 69)

Zierler, N. (1959). Linear recurring sequences. *J. Soc. Ind. Appl. Math. 7*(1), 31–48. (Cit. on pp. 18, 24).

Annexe A

Appendices

A.1 Table of Mlbs periodicity and set sizes

The periodicity of an Maximum Length Binary Sequence (Mlbs) is given by $N = 2^n - 1$, where n is the number of bits in the linear feedback shift registers (Lfsr) from which the Mlbs was generated (see Section 2.3.4). The number of independent Mlbs's M_n, including the mirror counterparts (see p. 23) for a period N is given by:

$$M_n = \frac{1}{n}\,\varphi(2^n - 1) = \frac{1}{n}\,\varphi(N) \tag{A.1.1}$$

(Barker, 1993), where $\varphi(x)$, the Euler's totient function, counts the number of positive integers less than or equal to x that are relatively prime to x (excluding the common divisor 1). Values of N and M_n for n between 3 and 30 are tabulated overleaf.

Table A.1: Table of MLBS periodicity and set sizes

n	N	M_n
3	7	2
4	15	2
5	31	6
6	63	6
7	127	18
8	255	16
9	511	48
10	1023	60
11	2047	176
12	4095	144
13	8191	630
14	16 383	756
15	32 767	1800
16	65 535	2048
17	131 071	7710
18	262 143	7776
19	524 287	27 594
20	1 048 575	24 000
21	2 097 151	84 672
22	4 194 303	120 032
23	8 388 607	356 960
24	16 777 215	276 480
25	33 554 431	1 296 000
26	67 108 863	1 719 900
27	134 217 727	4 202 496
28	268 435 455	4 741 632
29	536 870 911	18 407 808
30	1 073 741 823	17 820 000
⋮	⋮	⋮
n	$2^n - 1$	$\frac{1}{n}\,\varphi(2^n - 1)$

A.2 The theoretical BLAs for discrete-time Volterra systems

In this section, the detailed steps in deriving the Best Linear Approximations (BLAs) for generalised discrete-time Volterra systems for white inputs with different input amplitude distributions are shown. The assumptions made are similar to those in Section 3.3, as follows:

1. The Volterra system is stable, causal, and time-invariant. By definition a Volterra system has the periodicity-invariant (Pi) property of Section 2.1.
2. Input u is spectrally white, stationary, known, has zero mean and persistently exciting.

The theory in this section applies to system in feedback, as long as the input-output relationship can be approximated to a good degree by a finite-order Volterra system, given the above assumptions are satisfied. Lastly, the approach here based on multiset theory was not adapted from any existing literature.

A.2.1 Arbitrary input case

Volterra kernels are symmetrical, meaning that the permutation of their dimension arguments gives the same numerical result. For example, $h(a,b,c) = h(b,a,c) = h(c,b,a)$. It is useful to bring in the multiset notation where the order of elements does not matter. A multiset is a generalisation of set theory where multiple member elements can take the same numerical value, unlike those in a traditional set where they must be unique. Let $\boldsymbol{\Xi}$, in bold type, be an order-irrelevant multiset with $q-1$ members containing the kernel argument such that $\boldsymbol{\Xi} = \{k_1, \dots, k_{q-1}\}$ and all elements are integers. Let $\mathcal{P}$ be a class containing all possible distinct multisets of $\boldsymbol{\Xi}$, i.e. all the combinations that can be made from $q-1$ members drawn from the integer population of $\mathbb{Q} = \{0, 1, 2, ..., N-1\}$. These combinations conform to the order irrelevance principle, that is, for example, the set $\{a, a, b\}$ is equal to $\{a, b, a\}$ and either of which (but not both) appears in $\mathcal{P}$ only once. Formally this is equivalent to collapsing the ordered $(q-1)$-tuples arising from the *Cartesian products* of $\mathbb{Q}$ to the power of $q-1$ (i.e. $\mathbb{Q}^{(q-1)}$) into order-irrelevant $(q-1)$-tuples.

Let $\boldsymbol{\xi} \subseteq \boldsymbol{\Xi}$ be a traditional set (i.e., not a multiset) populated by unique elements of the multiset $\boldsymbol{\Xi}$. Finally, denote the function $\mathbb{1}_{\boldsymbol{A}}(x)$, called the *multiplicity function* for multisets, to count the number of occurrences of an element x in the multiset $\boldsymbol{A}$. Considering only zero-mean input signals, only contributions from *odd*

Volterra kernels are non-trivial (not equal to zero). An odd degree q^{th} kernel means the number of member elements in $\boldsymbol{\Xi}$, i.e. the cardinality $|\boldsymbol{\Xi}| = q - 1$ is even.

Example To better explain the notations introduced, consider a simple case with a discrete-time 3^{rd} order Volterra kernel $h[a, b, c]$. Arbitrarily let the length of the kernel $N = 3$, such that the kernel now has dimension $N \times q$ and has elements from $h[0, 0, 0]$ through $h[0, 0, 1], h[0, 0, 2], h[0, 1, 0]$ ... to $h[2, 2, 2]$. In this case, the multiset $\boldsymbol{\Xi}$ has cardinality $|\boldsymbol{\Xi}| = 3 - 1 = 2$. The class $\mathcal{P}$ contains all possible distinct multisets of $\boldsymbol{\Xi}$ such that $\mathcal{P} = \big\{\!\!\big\{\{0, 0\}, \{0, 1\}, \{0, 2\}, \{1, 1\}, \{1, 2\}, \{2, 2\}\big\}\!\!\big\}$. Note the aforementioned order irrelevance principle means the 2-tuples $\{a, b\} = \{b, a\}$ are counted as a single member multiset in $\mathcal{P}$.

The traditional sets $\boldsymbol{\xi} \subseteq \boldsymbol{\Xi}$ contain unique elements of their related $\boldsymbol{\Xi}$, so that if $\boldsymbol{\Xi} = \{2, 2\} \iff \boldsymbol{\xi} = \{2\}$ and if $\boldsymbol{\Xi} = \{0, 2\} \iff \boldsymbol{\xi} = \{0, 2\}$. Using this notation with respect to (3.36), it is sufficient to represent the Volterra kernel by:

$$\binom{q}{\{\mathbb{1}_{\boldsymbol{\Xi}}(\forall p \neq r \in \boldsymbol{\xi})\}, \mathbb{1}_{\boldsymbol{\Xi}}(r) + 1} h[\boldsymbol{\Xi}, r] \qquad \forall \boldsymbol{\Xi} \in \mathcal{P} \tag{A.2.1}$$

where $\binom{q}{\{\mathbb{1}_{\boldsymbol{\Xi}}(\forall p \in \boldsymbol{\xi})\}, \mathbb{1}_{\boldsymbol{\Xi}}(r)+1}$ is the *multinomial coefficient* arising from the number of ways of permuting the arguments in the Volterra kernel $h[\boldsymbol{\Xi}, r]$, i.e. $\binom{n}{k_1, k_2, \ldots, k_m} \triangleq \frac{n!}{k_1! k_2! \ldots k_m!}$.

Again with respect to (3.36), with $q = 3$, consider the fact that the Volterra kernel slices $h[0, 0, 1] = h[0, 1, 0] = h[1, 0, 0]$ are counted three times resulting in the contribution $\big(h[0, 0, 1] + h[0, 1, 0] + h[1, 0, 0]\big)\mathrm{E}\big[\!\big[u^2[t - 0]u^2[t - 1]\big]\!\big] = 3h[0, 1, 0]\,\mathfrak{M}_n^2$, where $\mathfrak{M}_n \triangleq \mathrm{E}[\![u^n]\!]$ uses the moment notation defined in Section 3.2. For the case of $r = 0$, the kernel combination is represented by $h[\boldsymbol{\Xi}, 0]$ where $\boldsymbol{\Xi} = \{0, 1\}$. Using (A.2.1), the contribution can be represented in multiset notation by writing:

$$\begin{aligned}
&\binom{3}{\{\mathbb{1}_{\boldsymbol{\Xi}}(1)\}, \mathbb{1}_{\boldsymbol{\Xi}}(0) + 1} h[\boldsymbol{\Xi}, r]\mathrm{E}\big[\!\big[u^2[t - 0]u^2[t - 1]\big]\!\big] \\
&= \binom{3}{1, (1 + 1)} h[0, 1, 0]\mathrm{E}\big[\!\big[u^2[t - 0]u^2[t - 1]\big]\!\big] \\
&= \frac{3!}{1!\,2!} h[0, 1, 0]\,\mathfrak{M}_n^2,
\end{aligned}$$

one obtains

$$= 3h[0, 1, 0]\,\mathfrak{M}_n^2$$

as required.

Following on from these, an alternative form of (3.36) may be established as:

$$g_{\text{BLA}}^{(q)}[r] = \sum_{\forall \boldsymbol{\Xi} \in \mathcal{P}} \binom{q}{\{\mathbb{1}_{\boldsymbol{\Xi}}(\forall p \neq r \in \boldsymbol{\xi})\}, \mathbb{1}_{\boldsymbol{\Xi}}(r)+1} h[\boldsymbol{\Xi}, r] \prod_{\forall p \in \boldsymbol{\xi} \subseteq \boldsymbol{\Xi}} \mathfrak{M}_{\mathbb{1}_{\boldsymbol{\Xi}}(p)} \mathfrak{M}_{\mathbb{1}_{\boldsymbol{\Xi}}(r)+1}. \tag{A.2.2}$$

While the multiset notation may seem needlessly complicated and artificial in the current form, this will prove useful for the simplification of the BLA expression for the Gaussian input case in the next section. For a specific example on obtaining the BLA of a nonlinear system excited by inputs with an arbitrary amplitude distribution with respect to its 5^{th} Volterra kernel, refer to the workings in deriving (A.3.11).

A.2.2 Gaussian input case

Separating the multinomial coefficient with the unordered n-tuple nature of multiset theory allows one to simplify (A.2.2) further for inputs sequences with a Gaussian amplitude distribution.

Let the number of unique elements not equal to r in $\boldsymbol{\Xi}$ be m, i.e. $m = |\{\boldsymbol{\xi} \backslash r\}|$. Substituting for the moment terms $\mathfrak{M}$ using Table 3.1 in (A.2.2) and dropping the subscript $\boldsymbol{\Xi}$ in $\mathbb{1}_{\boldsymbol{\Xi}}$ for convenience gives:

$$\begin{aligned} g_{\text{BLA}}^{(q)}[r] &= \sum_{\forall \boldsymbol{\Xi} \in \mathcal{P}} \binom{q}{\mathbb{1}(k_1), ..., \mathbb{1}(k_m), [\mathbb{1}(r)+1]} \mathfrak{M}_{\mathbb{1}(k_1)} \cdots \mathfrak{M}_{\mathbb{1}(k_m)} \mathfrak{M}_{[\mathbb{1}(r)+1]} h(\boldsymbol{\Xi}, r) \\ &= \sum_{\forall \boldsymbol{\Xi} \in \mathcal{P}} \frac{q! \, [\mathbb{1}(r)+1]!! \prod_{i=1}^{m} (\mathbb{1}(k_i)-1)!!}{[\mathbb{1}(r)+1]! \prod_{i=1}^{m} \mathbb{1}(k_i)!} h[\boldsymbol{\Xi}, r] \end{aligned} \tag{A.2.3}$$

To simplify the above equation, first note that double factorials can be expressed in terms of ordinary factorials by using the identity:

$$(2k-1)!! \equiv \frac{(2k)!}{2^k (k!)}. \tag{A.2.4}$$

Using this to replace the double factorials in (A.2.3) and cancelling terms give:

$$\begin{aligned} g^{(q)}_{\text{BLA}}[r] &= \sum_{\forall \boldsymbol{\Xi} \in \mathcal{P}} \frac{q!\,[\mathbb{1}(r)+2]!\,\cancel{\prod_{i=1}^{m}(\mathbb{1}(k_i))!}}{[\mathbb{1}(r)+1]!\prod_{i=1}^{m}\left[\frac{1}{2}\mathbb{1}(k_i)!\right]2^{\frac{\mathbb{1}(k_i)}{2}}\cancel{\prod_{i=1}^{m}\mathbb{1}(k_i)!}} h[\boldsymbol{\Xi}, r] \\ &= \sum_{\forall \boldsymbol{\Xi} \in \mathcal{P}} \frac{q!\,[\mathbb{1}(r)+2]!}{\cancel{[\mathbb{1}(r)+1]!}\left[\frac{\mathbb{1}(r)+2}{2}\right]!\,2^{\frac{\mathbb{1}(r)+2}{2}}\prod_{i=1}^{m}\left[\frac{1}{2}\mathbb{1}(k_i)\right]!\,2^{\frac{\mathbb{1}(k_i)}{2}}} h[\boldsymbol{\Xi}, r] \\ &= \sum_{\forall \boldsymbol{\Xi} \in \mathcal{P}} \frac{q!\,\cancel{[\mathbb{1}(r)+2]}}{\left[\frac{\mathbb{1}(r)+2}{2}\right]!\,2^{\frac{\mathbb{1}(r)+2}{2}}\prod_{i=1}^{m}\left[\frac{1}{2}\mathbb{1}(k_i)\right]!\,2^{\frac{\mathbb{1}(k_i)}{2}}} h[\boldsymbol{\Xi}, r] \\ &= \sum_{\forall \boldsymbol{\Xi} \in \mathcal{P}} \frac{q!}{\prod_{\boldsymbol{\xi} \subset \boldsymbol{\Xi}}\left[\frac{1}{2}\mathbb{1}_{\boldsymbol{\Xi}}(\boldsymbol{\xi})\right]!\,2^{\frac{\mathbb{1}(\boldsymbol{\xi})}{2}}} h(\boldsymbol{\Xi}, r). \end{aligned} \tag{A.2.5}$$

Also, a variant of the double factorial identity of (A.2.4) can be written as:

$$(2k+1)!! \equiv \frac{(2k+1)!}{2^k(k!)}$$

so that:

$$(2k+1)! \equiv (2k+1)!!\,2^k(k!). \tag{A.2.6}$$

By making a substitution of $q!$ in (A.2.5) for $q!!$ using (A.2.6), (A.2.5) can be further simplified to:

$$g^{(q)}_{\text{BLA}}[r] = \sum_{\forall \boldsymbol{\Xi} \in \mathcal{P}} \frac{q!!\,2^{\frac{q-1}{2}}\left(\frac{q-1}{2}\right)!}{\prod_{\boldsymbol{\xi} \subset \boldsymbol{\Xi}}\left[\frac{1}{2}\mathbb{1}_{\boldsymbol{\Xi}}(\boldsymbol{\xi})\right]!\,2^{\frac{\mathbb{1}(\boldsymbol{\xi})}{2}}} h[\boldsymbol{\Xi}, r]. \tag{A.2.7}$$

Observe that:

$$\begin{aligned} \frac{2^{\frac{q-1}{2}}}{\prod_{\boldsymbol{\xi} \subset \boldsymbol{\Xi}} 2^{\frac{\mathbb{1}(\boldsymbol{\xi})}{2}}} &= \frac{2^{\frac{q-1}{2}}}{2^{\sum \frac{\mathbb{1}(\boldsymbol{\xi})}{2}}} \\ &= \frac{2^{(q-1)}}{2^{\sum \mathbb{1}_{\boldsymbol{\Xi}}(\boldsymbol{\xi})}} \end{aligned} \tag{A.2.8}$$

and notice $\sum \mathbb{1}_{\boldsymbol{\Xi}}(\boldsymbol{\xi})$ represents the sum of the number of all elements in $\boldsymbol{\Xi}$. This is the cardinality $|\boldsymbol{\Xi}|$ and is by definition equal to $q-1$; hence the numerator and denominator of (A.2.8) cancel out. Using the same simplification in (A.2.7) yields:

$$g^{(q)}_{\text{BLA}}[r] = \sum_{\forall \boldsymbol{\Xi} \in \mathcal{P}} \frac{q!!\left(\frac{q-1}{2}\right)!}{\prod_{\boldsymbol{\xi} \subset \boldsymbol{\Xi}}\left[\frac{1}{2}\mathbb{1}_{\boldsymbol{\Xi}}(\boldsymbol{\xi})\right]!} h[\boldsymbol{\Xi}, r].$$

Expressing the factorials as a multinomial coefficient term gives:

$$= q!! \sum_{\forall \boldsymbol{\Xi} \in \mathcal{P}} \binom{\frac{q-1}{2}}{\{\frac{1}{2}\mathbb{1}_{\boldsymbol{\Xi}}(\forall p \in \boldsymbol{\xi})\}} h[\boldsymbol{\Xi}, r]. \tag{A.2.9}$$

The term responsible for the moments $q!!$ is decoupled and one is left only with a multinomial coefficient within the summation. The multinomial coefficient here corresponds to counting the permutations of the Volterra kernel arguments as pairs (see the next section), effectively treating each paired argument as single unified entities. Because of this, one can now replace the set notation with multi-dimensional summations and write, for zero-mean Gaussian inputs:

$$\begin{aligned} g^{(q)}_{\text{BLA}}[r] &= q!! \sum_{k_1}^{\infty} \cdots \sum_{k_p}^{\infty} h^{(q)}\left[k_1, k_1, k_2, k_2, \dots, k_p, k_p, r\right] \\ &= \mathfrak{M}_{(q+1)} \sum_{k_1}^{\infty} \cdots \sum_{k_p}^{\infty} h^{(q)}\Big[\overbrace{k_1, k_1}^{\text{pairing}}, k_2, k_2, \dots, k_p, \underbrace{k_p, r}_{q-1 \text{ terms, } p \text{ pairs}}\Big] \end{aligned} \tag{A.2.10}$$

where q is odd and $p = (q-1)/2$.

Non-triviality arising from the pairing of the Volterra kernel dimension arguments

This pair-wise dimension argument arrangement in (A.2.10) arises from the fact that, considering zero-mean input sequences in which all of their *odd* order higher moments are zero; for (3.36) to be non-trivial, even order higher moment terms must be constructed from the combination of dimension arguments k_α of (3.36) in *pairs*, where a pair refers to any two arguments in the Volterra kernel numerically equal to each other (Schetzen, 1980/2006). As such, the Volterra kernel is non-trivial only when $\mathbb{1}_{\boldsymbol{\Xi}}(x) \in 2\mathbb{N}_0 \, \forall x \in \boldsymbol{\xi}$ and it is only necessary to sum contributions where this criterion applies. Note that pairing does not preclude the occurrence of quadruplets, sextuplets, etc. if the degree of the kernel is sufficient. The only exception is in the element of interest r, where there must be an odd number of them occurring in the argument for the BLA to be non-trivial, due to the cross-correlation operation introducing an extra '$u[t-r]$' term in (3.36).

Another explanation for the unique ability of Gaussian sequences to decompose a Volterra summation of the form in (3.36) into paired argument form in (A.2.10) is through the Isserlis' theorem (Isserlis, 1916, 1918). The theorem states that, for $x \sim \mathcal{N}$:

$$\mathrm{E}[\![x_1 x_2 \cdots x_{2n}]\!] = \sum \prod \mathrm{E}[\![x_a x_b]\!] \tag{A.2.11}$$

where the notation $\sum\prod$ signifies the sum over all distinct ways of partitioning $x_1 x_2 \cdots x_{2n}$ into pairs (Michalowicz, Nichols, Bucholtz & Olson, 2009). This is compatible with how non-zero terms arise from the BLA as explained in this section.

A similar pairing characteristic arises when the analysis is performed in the frequency domain (see Pintelon & Schoukens, 2012, app. 3.B).

A.2.3 Binary input case

The simple result in (A.2.10) is valid if and only if the input has a Gaussian amplitude distribution as the simplification relies on the specific combinatorial nature of the Gaussian moments. For other input sequences such as binary sequences, it is not possible to decouple the moment terms (all of which equal unity) with the multinomial coefficients. A correction factor is required to compensate for the combinatorial deficiency when summing paired arguments in order to match the exact number occurrence of terms with an argument pattern of the form $h\big[k_1, k_1, k_2, k_2, \ldots, k_p, k_p, r\big]$, to those naturally arise from of the general summation operation of $h\big[k_1, k_2, \ldots, k_{q-1}, r\big]$ in (3.36).

The following example illustrates how differences arise between Gaussian and other sequences in computing the theoretical BLA with a 7^{th} order Volterra kernel; the relatively high kernel degree showcases the effect of various argument combinations on the constant of proportionality.

Example Consider a BLA with a seventh Volterra kernel contribution $h^{(7)}$ only with input denoted as u. Using (3.36), part of the BLA includes the summation term of the form:

$$S_{2,4,1} = \sum_{k_1=0}^{\infty}\sum_{k_2=0}^{\infty} h^{(7)}\left[k_1, k_1, k_2, k_2, k_2, k_2, r\right] u^2[t-k_1]u^4[t-k_2]u^2[t-r]\Big|_{k_1\neq k_2}.$$

Within the summation of $S_{2,4,1}$, terms can only have the following three forms:

$$\begin{cases} \binom{7}{2,4,1}\, h^{(7)}\left[k_1, k_1, k_2, k_2, k_2, k_2, r\right] \mathfrak{M}_2^2\, \mathfrak{M}_4 & \text{for } k_1 \neq k_2 \neq r \\ \binom{7}{4,3}\, h^{(7)}\left[k_1, k_1, k_2, k_2, k_2, k_2, r\right] \mathfrak{M}_4^2 & \text{for } k_2 \neq r, k_1 = r \\ \binom{7}{2,5}\, h^{(7)}\left[k_1, k_1, k_2, k_2, k_2, k_2, r\right] \mathfrak{M}_2\, \mathfrak{M}_6 & \text{for } k_1 \neq r, k_2 = r \end{cases}$$

$$= \begin{cases} 105\; h^{(7)}\left[k_1, k_1, k_2, k_2, k_2, k_2, r\right] \mathfrak{M}_2^2\, \mathfrak{M}_4 \\ 35\; h^{(7)}\left[r, r, k_2, k_2, k_2, k_2, r\right] \mathfrak{M}_4^2 \\ 21\; h^{(7)}\left[k_1, k_1, r, r, r, r, r\right] \mathfrak{M}_2\, \mathfrak{M}_6\,. \end{cases} \tag{A.2.12}$$

If $u \sim \mathcal{N}(0,1)$, using Table 3.1, $\mathfrak{M}_2^2\,\mathfrak{M}_4 = 3$, $\mathfrak{M}_4^2 = 9$ and $\mathfrak{M}_2\,\mathfrak{M}_6 = 15$. Substituting these moment terms into (A.2.12) one obtains:

$$= \begin{cases} 315\ h^{(7)}\left[k_1, k_1, k_2, k_2, k_2, k_2, r\right] & \text{for } k_1 \neq k_2 \neq r \\ 315\ h^{(7)}\left[r, r, k_2, k_2, k_2, k_2, r\right] & \text{for } k_2 \neq r \\ 315\ h^{(7)}\left[k_1, k_1, r, r, r, r, r\right] & \text{for } k_1 \neq r. \end{cases} \tag{A.2.13}$$

Observe that the constants of proportionality for all three forms are equal to 315. This is not the case if $\langle u \rangle$ is non-Gaussian. For instance, if $\langle u \rangle$ is a uniform zero-mean random binary sequence with all higher order moments equal to unity, the contributions from different Volterra kernel slices probed at time element of interest r are summed in different proportions. This disproportionality results in a bias.

It is possible to rewrite (3.36) to consider only the summation of non-trivial terms for binary sequences by using the multiset notation, similar to steps performed in obtaining (A.2.2). This results in:

$$g^{(q)}_{\mathcal{U}_2\,\mathrm{BLA}}[r] = \sum_{\forall \boldsymbol{\Xi} \in \mathcal{P}} \binom{\frac{q-1}{2}}{\{\frac{1}{2}\mathbb{1}_{\boldsymbol{\Xi}}(\forall p \in \boldsymbol{\xi})\}} \frac{1}{[\mathbb{1}_{\boldsymbol{\Xi}}(r)+1]!! \prod_{i=1}^{m} (\mathbb{1}_{\boldsymbol{\Xi}}(k_i)-1)!!} h[\boldsymbol{\Xi}, r] \tag{A.2.14}$$

$$= \sum_{k_1}^{\infty} \cdots \sum_{k_p}^{\infty} \gamma(\boldsymbol{\Xi}, \boldsymbol{r}) h^{(q)}\left[k_1, k_1, k_2, k_2, \ldots, k_p, k_p, r\right]. \tag{A.2.15}$$

Here $\gamma(\boldsymbol{\Xi}, \boldsymbol{r})$ is a correction term related to the reciprocal of the products of Gaussian moments, with its value dependent on which and how many of the k_i terms are equal to each other and which of them are equal to r.

For a specific example on obtaining the BLA of a nonlinear system excited by zero-mean binary inputs with respect to its 5^{th} kernel, refer to the workings in deriving (A.3.6) in Appendix A.3.

A.3 The BLAs of a Wiener system with a pure quintic nonlinearity

A.3.1 Gaussian input case

Using (A.2.10), the BLA of a nonlinear system described by a 5^{th} degree Volterra kernel excited by zero-mean Gaussian inputs with normalised power is given by:

$$\begin{aligned} g^{(5)}_{\mathcal{N}\,\text{BLA}}[r] &= 5!! \sum_{k_1}^{\infty}\sum_{k_2}^{\infty} h^{(q)}\left[k_1, k_1, k_2, k_2, r\right] \\ &= 15 \sum_{k_1}^{\infty}\sum_{k_2}^{\infty} h^{(q)}\left[k_1, k_1, k_2, k_2, r\right]. \end{aligned} \tag{A.3.1}$$

If the nonlinear system is a Wiener system constructed from a linearity with impulse response function (IRF) of $g[k]$ followed by a pure quintic nonlinearity, its Volterra kernel $h[a, b, c, d, e]$ collapses into the product $g[a]g[b]g[c]g[d]g[e]$. In such a case, (A.3.1) is simplified to:

$$\begin{aligned} g^{(5)}_{\mathcal{N}\,\text{BLA}}[r] &= 15 \sum_{k_1}^{\infty}\sum_{k_2}^{\infty} g^2[k_1]g^2[k_2]g[r] \\ &= 15g[r] \sum_{k_1}^{\infty}\sum_{k_2}^{\infty} g^2[k_1]g^2[k_2] \\ &= 15g[r] \left(\sum_{k=0}^{\infty} g^2[k]\right)^2. \end{aligned} \tag{A.3.2}$$

A.3.2 Binary input case

With respect to zero-mean binary inputs, using (A.2.14), one can deduce that a 5^{th} degree Volterra series contains the following forms of non-trivial (non-zero) contributions:

$$\begin{cases} \binom{5}{2,2,1}\, h^{(5)}\left[k_1, k_1, k_2, k_2, r\right] & \text{for } k_1 \neq k_2 \neq r \\ \binom{5}{2,3}\, h^{(5)}\left[k, k, r, r, r\right] & \text{for } k_1 \neq r, k_2 = r \\ \binom{5}{4,1}\, h^{(5)}\left[k, k, k, k, r\right] & \text{for } k_1 = k_2 \neq r \\ \binom{5}{5}\, h^{(5)}\left[r, r, r, r, r\right] & \text{for } k_1 = k_2 = r \end{cases}$$

$$\begin{cases} 15h^{(5)}\left[k_1, k_1, k_2, k_2, r\right] \\ 10h^{(5)}\left[k, k, r, r, r\right] \\ 5h^{(5)}\left[k, k, k, k, r\right] \\ h^{(5)}\left[r, r, r, r, r\right]. \end{cases} \tag{A.3.3}$$

The binary BLA is then given by:

$$g^{(5)}_{\mathcal{U}_2\,\mathrm{BLA}}[r] = 15 \underbrace{\sum_{k_1=0}^{\infty}\sum_{k_2=0}^{\infty}}_{k_1\neq k_2\neq r} h^{(5)}\left[k_1, k_1, k_2, k_2, r\right] \tag{A.3.4a}$$

$$+ 10\sum_{\substack{k=0\\k\neq r}}^{\infty} h^{(5)}\left[k, k, r, r, r\right] \tag{A.3.4b}$$

$$+ 5\sum_{\substack{k=0\\k\neq r}}^{\infty} h^{(5)}\left[k, k, k, k, r\right] \tag{A.3.4c}$$

$$+ h^{(5)}\left[r, r, r, r, r\right]. \tag{A.3.4d}$$

Looking at (A.3.4a) (the first term), it is beneficial to relax the requirement of $k_1 \neq k_2 \neq r$ within the double summations. Before dealing with the first term, which incorporates the other terms within the summations, it is easier to firstly consider the relaxation of the requirements of $k \neq r$ for (A.3.4b) and (A.3.4c). One may define $S_{2,3}$ and $S_{4,1}$, respectively, for (A.3.4b) and (A.3.4c) as:

$$\begin{aligned} S_{2,3} &\triangleq \sum_{k=0}^{\infty} h^{(5)}\left[k, k, r, r, r\right] \\ &= \sum_{\substack{k=0\\k\neq r}}^{\infty} h^{(5)}\left[k, k, r, r, r\right] + h^{(5)}\left[r, r, r, r, r\right] \end{aligned}$$

$$\therefore \sum_{\substack{k=0\\k\neq r}}^{\infty} h^{(5)}\left[k, k, r, r, r\right] = S_{2,3} - h^{(5)}\left[r, r, r, r, r\right] \tag{A.3.5a}$$

and

$$\begin{aligned} S_{4,1} &\triangleq \sum_{k=0}^{\infty} h^{(5)}\left[k, k, k, k, r\right] \\ &= \sum_{\substack{k=0\\k\neq r}}^{\infty} h^{(5)}\left[k, k, k, k, r\right] + h^{(5)}\left[r, r, r, r, r\right] \end{aligned}$$

$$\therefore \sum_{\substack{k=0 \\ k\neq r}}^{\infty} h^{(5)}\left[k,k,k,k,r\right] = S_{4,1} - h^{(5)}\left[r,r,r,r,r\right]. \tag{A.3.5b}$$

Similarly, consider the full unrestricted double sum denoted by $S_{2,2,1}$ and writing it in terms of (A.3.4a):

$$\begin{aligned} S_{2,2,1} &\triangleq \sum_{k_1=0}^{\infty}\sum_{k_2=0}^{\infty} h^{(5)}\left[k_1,k_1,k_2,k_2,r\right] \\ &= \underbrace{\sum_{k_1=0}^{\infty}\sum_{k_2=0}^{\infty}}_{k_1\neq k_2\neq r} h^{(5)}\left[k_1,k_1,k_2,k_2,r\right] \\ &\quad + \binom{2}{1}\sum_{\substack{k=0 \\ k\neq r}}^{\infty} h^{(5)}\left[k,k,r,r,r\right] \\ &\quad + \sum_{\substack{k=0 \\ k\neq r}}^{\infty} h^{(5)}\left[k,k,k,k,r\right]. \end{aligned}$$

Substituting (A.3.5a) and (A.3.5b) into the equation yields:

$$\begin{aligned} S_{2,2,1} &= \underbrace{\sum_{k_1=0}^{\infty}\sum_{k_2=0}^{\infty}}_{k_1\neq k_2\neq r} h^{(5)}\left[k_1,k_1,k_2,k_2,r\right] \\ &\quad + 2S_{2,3} - 2h^{(5)}\left[r,r,r,r,r\right] \\ &\quad + S_{4,1} - h^{(5)}\left[r,r,r,r,r\right] \\ &= \underbrace{\sum_{k_1=0}^{\infty}\sum_{k_2=0}^{\infty}}_{k_1\neq k_2\neq r} h^{(5)}\left[k_1,k_1,k_2,k_2,r\right] \\ &\quad + 2S_{2,3} + S_{4,1} - 3h^{(5)}\left[r,r,r,r,r\right] \end{aligned}$$

$$\therefore \underbrace{\sum_{k_1=0}^{\infty}\sum_{k_2=0}^{\infty}}_{k_1\neq k_2\neq r} h^{(5)}\left[k_1,k_1,k_2,k_2,r\right] = S_{2,2,1} - 2S_{2,3} - S_{4,1} + 3h^{(5)}\left[r,r,r,r,r\right]. \tag{A.3.5c}$$

Finally, substitute (A.3.5a), (A.3.5b) and (A.3.5c) into (A.3.4) one obtains the BLA of a Wiener system with a pure quintic nonlinearity excited by zero-mean binary

inputs as:

$$\begin{aligned} g^{(5)}_{\mathcal{U}_2\,\mathrm{BLA}}[r] &= 15S_{2,2,1} - 30S_{2,3} - 15S_{4,1} + 45h^{(5)}\,[r,r,r,r,r] \\ &\quad + 10S_{2,3} - 10h^{(5)}\,[r,r,r,r,r] \\ &\quad + 5S_{4,1} - 5h^{(5)}\,[r,r,r,r,r] \\ &\quad + h^{(5)}\,[r,r,r,r,r] \\ &= 15S_{2,2,1} - 20S_{2,3} - 10S_{4,1} + 16h^{(5)}\,[r,r,r,r,r] \\ g^{(5)}_{\mathcal{U}_2\,\mathrm{BLA}}[r] &= 15\sum_{k_1=0}^{\infty}\sum_{k_2=0}^{\infty} h^{(5)}\,[k_1,k_1,k_2,k_2,r] - 20\sum_{k=0}^{\infty} h^{(5)}\,[k,k,r,r,r] \\ &\quad - 10\sum_{k=0}^{\infty} h^{(5)}\,[k,k,k,k,r] + 16h^{(5)}\,[r,r,r,r,r]. \end{aligned} \tag{A.3.6}$$

One can also write (A.3.6) in terms of the Gaussian case of (A.3.1), resulting in:

$$\begin{aligned} g^{(5)}_{\mathcal{U}_2\,\mathrm{BLA}}[r] &= g^{(5)}_{\mathcal{N}\,\mathrm{BLA}}[r] - 20\sum_{k=0}^{\infty} h^{(5)}\,[k,k,r,r,r] \\ &\quad - 10\sum_{k=0}^{\infty} h^{(5)}\,[k,k,k,k,r] + 16h^{(5)}\,[r,r,r,r,r]. \end{aligned} \tag{A.3.7}$$

Again, if the nonlinear system is a Wiener system constructed from a linearity with IRF of $g[k]$ followed by a pure quintic nonlinearity, its Volterra kernel $h[a,b,c,d,e]$ collapses into the product $g[a]g[b]g[c]g[d]g[e]$. In such a case, (A.3.7) is simplified to:

$$\begin{aligned} g^{(5)}_{\mathcal{U}_2\,\mathrm{BLA}}[r] &= g^{(5)}_{\mathcal{N}\,\mathrm{BLA}}[r] - 20g^3[r]\sum_{k=0}^{\infty} g^2[k] \\ &\quad - 10g[r]\sum_{k=0}^{\infty} g^4[k] + 16g^5[r]. \end{aligned} \tag{A.3.8}$$

A.3.3 Arbitrary input case

For any zero-mean arbitrary input, similar to the steps performed in deriving (A.3.3), there are non-trivial contributions of the Volterra series in the form of:

$$\begin{cases} \binom{5}{2,2,1}\, h^{(5)}\,[k_1,k_1,k_2,k_2,r]\,(\mathfrak{M}_2)^3 & \text{for } k_1 \neq k_2 \neq r \\ \binom{5}{2,3}\, h^{(5)}\,[k,k,r,r,r]\,\mathfrak{M}_2\,\mathfrak{M}_4 & \text{for } k_1 \neq r, k_2 = r \\ \binom{5}{4,1}\, h^{(5)}\,[k,k,k,k,r]\,\mathfrak{M}_2\,\mathfrak{M}_4 & \text{for } k_1 = k_2 \neq r \\ \binom{5}{5}\, h^{(5)}\,[r,r,r,r,r]\,\mathfrak{M}_6 & \text{for } k_1 = k_2 = r \end{cases} \tag{A.3.9}$$

where $\mathfrak{M}_n \triangleq \mathrm{E}[\![u^n]\!]$ is the moment notation defined in Section 3.2.

Recall from (3.27) that the definition of the non-Gaussian moment correction terms is $\delta_n \triangleq \mathrm{E}\left[\!\left[u^n[k]\right]\!\right] - (n-1)!!\,\sigma^n$ for even n, where $u^n[k]$ is some zero-mean arbitrary sequence. Normalising the power of the sequences such that $\sigma^2 = \mathfrak{M}_2 = \mathrm{E}\left[\!\left[u^2[k]\right]\!\right] = 1$ and hence $\delta_2 = 0$, one can write the contributions to BLA of this arbitrary sequence in terms of Gaussian moment by applying (A.3.9) for both Gaussian and arbitrary signals as:

$$\begin{aligned}
g^{(5)}_{\mathcal{X}\,\mathrm{BLA}} &= g^{(5)}_{\mathcal{N}\,\mathrm{BLA}} + g^{(5)}_{\mathcal{X}\,\mathrm{BLA}} - g^{(5)}_{\mathcal{N}\,\mathrm{BLA}} \\
&= g^{(5)}_{\mathcal{N}\,\mathrm{BLA}} + (15-15)\underbrace{\sum_{k_1=0}^{\infty}\sum_{k_2=0}^{\infty}}_{k_1\neq k_2\neq r} h^{(5)}\left[k_1,k_1,k_2,k_2,r\right] \\
&\quad + 10\delta_4 \sum_{\substack{k=0\\k\neq r}}^{\infty} h^{(5)}\left[k,k,r,r,r\right] \\
&\quad + 5\delta_4 \sum_{\substack{k=0\\k\neq r}}^{\infty} h^{(5)}\left[k,k,k,k,r\right] \\
&\quad + \delta_6 h^{(5)}\left[r,r,r,r,r\right] \\
g^{(5)}_{\mathcal{X}\,\mathrm{BLA}} &= g^{(5)}_{\mathcal{N}\,\mathrm{BLA}} + 10\delta_4 \sum_{k=0}^{\infty} h^{(5)}\left[k,k,r,r,r\right] \\
&\quad + 5\delta_4 \sum_{k=0}^{\infty} h^{(5)}\left[k,k,k,k,r\right] \\
&\quad + (\delta_6 - 15\delta_4) h^{(5)}\left[r,r,r,r,r\right].
\end{aligned} \tag{A.3.10}$$

Finally, analogous to (A.3.8), if the nonlinear system is a Wiener system with IRF of $g[k]$ followed by a pure quintic nonlinearity, (A.3.10) becomes:

$$\begin{aligned}
g_{\mathcal{X}\,\mathrm{BLA}}[r] &= g_{\mathcal{N}\,\mathrm{BLA}}[r] + 10\delta_4 g^3[r] \sum_{k=0}^{\infty} g^2[k] \\
&\quad + 5\delta_4 g[r] \sum_{k=0}^{\infty} g^4[k] + (\delta_6 - 15\delta_4) g^5[r].
\end{aligned} \tag{A.3.11}$$

Using the definition of δ_n for a binary input with power of unity, $\delta_4 = 1-(4-1)!! = 1-3 = -2$ and $\delta_6 = 1-(6-1)!! = 1-15 = -14$. Substituting these into (A.3.11) one obtains again (A.3.8), as required.

A.4 Settling time for first and second order systems

This section explains the definition of the 1% settling time in the x-axis labels in Fig. 3.7 and 3.9 of Section 3.5.2. The unit IRF is the response of a system subjected to an ideal Dirac delta function as input $u(t > 0) = \delta(t)$. For a first order linear dynamical system the unit IRF is given by:

$$y(t) = e^{-\frac{t}{\tau}} \tag{A.4.1}$$

where τ is the time constant. There is no universal definition for the length of an exponentially decaying IRF. To enable comparison with a second order system where the definition of a time constant is non-existent, Section 3.5.2 takes the settling time to be 1% of the initial value at $t = 0$.

First order system

The time taken for the first order unit IRF to reach 1% of its initial value is:

$$\begin{aligned} e^{-\frac{t}{\tau}} &= 0.01 \\ t &= -\tau \ln 0.01 = 4.605\tau. \end{aligned} \tag{A.4.2}$$

Some may prefer to define the settling time as 5 time constants.

Second order system

The unit IRF for an under-damped ($0 < \zeta < 1$) second order linear dynamical system is given by:

$$\begin{aligned} y(t) &= \frac{1}{\omega_\mathrm{n}\sqrt{1-\zeta^2}}\, e^{-\zeta\omega_\mathrm{n}t} \sin\left(\omega_\mathrm{n}t\sqrt{1-\zeta^2}\right) \\ &= Ce^{-\zeta\omega_\mathrm{n}t} \sin\left(\omega_\mathrm{n}t\sqrt{1-\zeta^2}\right) \end{aligned} \tag{A.4.3}$$

where ζ is the damping coefficient, ω_n is the undamped natural frequency and C is a scalar constant. To enable fair comparison with the first order case, the oscillatory sinusoidal term of this second order IRF is ignored. The scalar constant does not affect the ratio of initial value to the 1% value. Therefore, the settling time hence depends only upon the exponent term $e^{-\zeta\omega_\mathrm{n}t}$, such that:

$$\begin{aligned} e^{-\zeta\omega_\mathrm{n}t} &= 0.01 \\ t &= \frac{-\ln 0.01}{\zeta\omega_\mathrm{n}}. \end{aligned} \tag{A.4.4}$$

The knowledge of $\zeta\omega_n$ may be extracted from the poles the second order system through workings detailed in Appendix A.5.

A.5 Obtaining ζ and ω_n from z-domain poles of a second order system

This section documents the steps in obtaining the damping coefficient ζ and undamped natural frequency ω_n from z-domain poles of a second order linear dynamical system, required in Section 3.5.2 of the main text. A continuous time second order linear dynamical system may be described by the s-domain representation of

$$s^2 + \zeta\omega_n s + \omega_n^2 = 0.$$

Solving for parameter s gives:

$$s = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2}. \tag{A.5.1}$$

It is possible to obtain the sampled z-domain representation by the relation $z \triangleq e^{sT}$ where T is the sampling interval. (A.5.1) can then be written as:

$$z = e^{-\zeta\omega_n T \pm j\omega_n T\sqrt{1-\zeta^2}}. \tag{A.5.2}$$

The sampling interval T, for simplicity, is taken as 1 in the main text.

Let poles in z-domain be defined by polar coordinate $e^{-r}(\cos\theta \pm j\sin\theta)$, by Euler's formula this is equivalent to $(e^{-r\pm j\theta})$. By definition for a given pole p, $z - p = 0$. Equating the poles in (A.5.2) and taking natural logarithm of both sides give:

$$-r \pm j\theta = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2}. \tag{A.5.3}$$

Equating real and imaginary parts yields:

$$\begin{cases} r = \zeta\omega_n \\ \theta = \omega_n\sqrt{1-\zeta^2}. \end{cases} \tag{A.5.4}$$

It can be seen that the product term $\zeta\omega_n$ of (A.4.4) is simply equal to r, the vector of independent variables tested in Section 3.5.2. For completeness, solving simultaneously for ζ and ω_n in terms of r and θ gives:

$$\begin{cases} \zeta = \dfrac{r}{\sqrt{r^2+\theta^2}} \\ \omega_n = \sqrt{r^2+\theta^2}. \end{cases} \tag{A.5.5}$$

Annexe B

MATLAB Program Codes

B.1 Program code to generate MLBS's

This section incorporates the MATLAB code used alongside with the package *prs* (Tan & Godfrey, 2002; Godfrey, Tan et al., 2005) to generate MLBS's (see 2.3.4).

```
function out=mlbs_genpoly(m)
%This is a function to generate a reduced set of primitive polynomials of a
    generating polynomial of degree m by eliminating mirror counterparts
%Call the primpoly function in prs package to obtain all primitive polynomials:
pr = primpoly(m,'all','nodisplay');
%Discard Least and Most Significant bit for mirror sequence elimination algorithm:
pr=(pr-2^m-1)/2;
%Eliminating mirror sequences speed up m-sequence generation later
prx=zeros(ceil(size(pr,1)/2),1); %Reduced primitive polynomial set
prx(1) = bin2dec(fliplr(dec2bin(pr(1),m-1))); %Reverse feedback taps
k=2;
for i=2:1:size(pr,1)
    mirror = bin2dec(fliplr(dec2bin(pr(i),m-1))); %Reverse feedback taps
    if mirror > pr(i) %Check if it is a mirror, binary number would be bigger
        prx(k)=mirror;
        k=k+1;
    end
end
prx=sort(prx);
out=dec2bin(prx*2+1+2^m); %add the MSB and LSB back
end
```

Listing B.1: mlbs_genpoly.m

```
function [out]=mlbs_gensequences(m)
%This is a function which generate a family of m-sequences with a common period N
    =2^m-1. The mirror counterparts of each sequence are omitted but can easily be
     obtained by time-reversing each m-sequence in the output. The output is a
    MATLAB structure with field 'config' containing the primitive polynomial as a
    binary string and 'sequence' containing the sequence as a vector.
primpoly = mlbs_genpoly(m); %Call the primitive polynomial helper function

N = size(primpoly,1); %Preallocate memory
MLBS_series(N,1) = struct('config', NaN(1,size(primpoly,2)), 'sequence',zeros(2^(
    size(primpoly,2)-1)-1,1,'int8')); %Preallocate memory
for i=1:N
MLBS_series(i).config = primpoly(i,:);
MLBS_series(i).sequence = poly_MLB(primpoly(i,:)); %Call the poly to MLB function
    in the prs package
end
out=MLB_series;
```

Listing B.2: mlbs_gensequences.m

B.2 Program code to generate random-phased multisines

This section incorporates the MATLAB code to generate random-phased multisines through a procedure similar to that in Schoukens, Pintelon and Rolain (2012, p. 38). The amplitude and phase distributions can be altered according to user needs. Example code here generates multisines with white power spectra with root mean squared (rms) power of unity and uniform random phase distribution within $\pm\pi$ radians.

```
function u_msine=msine_gensequences(N,M)
% Generate M discrete-time random phased multisines of period N with flat power
    spectra
    A = ones(N,M); %M amplitude vectors of length N
    phi = rand(N,M)*2*pi+pi; %Phasor of random phase multisine with uniform phase
        distribution between -pi and +pi
    U = A.*exp((1i*phi)); %Generate the multisines' frequency spectra
    %Only the first half of the spectrum is correct, the 2nd half represents the
        negative frequencies hence requires complex conjugate folding. This is
        automatically taken care of by the 'symmetric' option of "ifft":
    u_msine = real(ifft(U,'symmetric')); %Generate the multisines
%    The above is equivalent to the following two commented lines:
%     U(1:ceil(N/2),:)=0;
%     u_msine2 = real(ifft(U)); (Schoukens 2012 p.38)
%    Factor of 2 in reference is unnecessary as signal is later normalised to rms
    = 1:
    for k=1:M
        u_msine(:,k)=u_msine(:,k)./std(u_msine(:,k)); %Normalise rms
    end
end
```

Listing B.3: msine_gensequences.m

B.3 Simulation experiment comparing stochastic and invariant spectrum signals

This section incorporates the MATLAB code used for the simulation experiment conducted in Section 4.3.

```
EXP=500; %500 experiments
N=2048; %Period of signals, arbitrary choice
%Test vector of M:
vecM=[1 2 3 4 5 6 7 10 11 12 13 14 15 16 20 21 22 23 24 25 30 31 32];
[b,a]=butter(6,0.5); %6th order butterworth lowpass, cutoff 0.5
MSE=zeros(length(vecM),2); %Grand total MSE, column 1 for multisine
mMSE=MSE; %Sub-experiment MSE
%Generate multisine
A= ones(N,max(vecM)); %Set all amplitudes to one, ideally only up to N/2 but this
    is a shortcut as ifft in line 16 take cares of this
for ex=1:EXP %Independent experiments
for m=1:length(vecM) %For different values of M (i.e. number of realisations)
    M=vecM(m);

    G=(freqz(b,a,N,'whole')*ones(1,M)); %Put real transfer function G0 in M
        columns

    phi = rand(N,M)*2*pi+pi; %Phasor of random phase multisine
    %Realise the multisine by using ifft trick, much quicker than adding
        individual sine waves in time domain. Going back to time domain with ifft
        and symmetric option fixes the incorrect folding of the fft spectrum
    U1 = fft(ifft(A(:,1:M).*exp((1i*phi)),'symmetric')*sqrt(N)); %Normalised power
        to 1
    u2=randn(N,M); %Gaussian noise inputs, also have power of 1
    U2=fft(u2); %Gaussian noise input in frequency domain

    Ns = fft(randn(N,M)*0.1); %AWGN noise in frequency domain, factor of 0.1
        result in variance 0.01, SNR -> 20dB
    Yhat1 = U1.*G+Ns; %System output for multisine, using frequency domain instead
        of convolution result in steady-state response (no transient)
    Yhat2 = U2.*G+Ns; %System output for Gaussian noise

    Ghat1= mean(Yhat1.*conj(U1),2)./mean(U1.*conj(U1),2); %Robust method
    Ghat2= mean(Yhat2.*conj(U2),2)./mean(U2.*conj(U2),2);

    mMSE(m,1) = sum(abs(Ghat1-G(:,1)).^2); %MSE error for each sub-experiment
    mMSE(m,2) = sum(abs(Ghat2-G(:,1)).^2); %MSE error for each sub-experiment
    MSE=MSE+mMSE/EXP; %Running average of MSE
end
display(int2str(ex));
end
```

Listing B.4: MSEvsM.m

B.4 General procedure for calculating the theoretical BLA for a Wiener system

This section incorporates the MATLAB code used for the simulation experiment conducted in Section 5.3.2, which may also be adapted as a general procedure for

calculating the theoretical BLA for inputs with an arbitrary amplitude distribution given a Wiener system with a polynomial nonlinearity of arbitrary order. Three files are listed here, firstly the main program `Theoretical_BLA.m` and two auxiliary functions `cumprodapdci.m` and `intpartitions.m`.

```
Trials = 5000; %Number of experiments for obtaining nonparametric BLA estimates
N=2048; %Periodicity of signals
P=2; %Number of periods - 1 for transients to decay
isLinear=false;
inputNoise=0;
outputNoise=0;

%True linearity impulse response ordinates
g_linearity = [1 0.6 0.1];  %Linearity IRF coeeficients used in chapter 6
l_linearity = length(g_linearity); %Length of linearity IRF

%Polynomial nonlinearity specificaton
Pn = [1 0 0 0 0 0 0 0 0 0]; %Coefficients in decreasing order (degree n....0)
Pn_rev = fliplr(Pn);
vecDeg = find(Pn_rev~=0)-1;
if isLinear
    nl = @(x) x;
else
    nl = @(x) polyval(Pn,x); %Nonlinearity
end

vecSettings = 1:5;
for setting=vecSettings %For all different input types defined below:
%inType -    1: discrete uniform distribution
%            2: user specified amplitude distribution
%    For inType = 2 only, specify:
%        inP - vector of probabilities corresponding to inL:
%        inL - vector of absolute values L signifying +/-L amplitude levels
%color - color of the line used in figure output
i=1;
structSettings(i).inType=1;
structSettings(i).name='Binary (Uniform)';
structSettings(i).color='g';
structSettings(i).uniform_a=0;
%==
i=i+1;
structSettings(i).inType=1;
structSettings(i).name='Ternary (Uniform)';
structSettings(i).color='c';
structSettings(i).uniform_a=1;
%==
i=i+1;
structSettings(i).inType=1;
structSettings(i).name='Quinary (Uniform)';
structSettings(i).color='r';
structSettings(i).uniform_a=2;
%==
i=i+1;
structSettings(i).inType=2;
structSettings(i).name='Ternary (Optimised)';
structSettings(i).color=[1 0.69 0.39];
structSettings(i).uniform_a=0;
structSettings(i).inP = [1/3]; %Probability of +/-level-L
structSettings(i).inL = [sqrt(3)]; %Amplitude level-L
%==
i=i+1;
structSettings(i).inType=2;
```

```
structSettings(i).name='Quadratic (Optimised)';
structSettings(i).color='m';
structSettings(i).uniform_a=0;
structSettings(i).inP = [(3-sqrt(6))/6 (3+sqrt(6))/6]; %Probability of +/-level-L
structSettings(i).inL = [sqrt(3+sqrt(6)) sqrt(3-sqrt(6))]; %Amplitude level-L
clear i

inType = structSettings(setting).inType; %Input type
switch inType
    case 1 %Arbitrary uniform discrete level signal for comparison with
        % Gaussian, amplitudes from -a to a in steps of 1, 0 for binary
        uniform_a = structSettings(setting).uniform_a;
    case 2 %User specified probability and amplitude levels
    inP = structSettings(setting).inP; %Probability of +/-level-L
    inL = structSettings(setting).inL; %Amplitude level-L
end

%Function of Higher order central moments definitions
switch inType
    %Moment needs to return 1 if input is 0 for the correct functioning of the
        routine.
    case 1 %discrete uniform distribution
    uniform_factor = sqrt((2*uniform_a+1)/(sum((1:uniform_a).^2)*2));
    fMomentsSingle=@(x) (x~=0)*(uniform_a~=0)*...
        ((sum(((1:uniform_a)*uniform_factor).^x))*2/(2*uniform_a+1))+...
        (uniform_a==0)+(x==0)-(uniform_a==0)*(x==0);
    case 2 %user specified random discrete sequence
    fMomentsSingle=@(x) (x~=0)*(sum((ceil(N*inP/2)).*((inL).^x)*2)/N)+(x==0);
end
fMoments = @(array) arrayfun(fMomentsSingle,array);
%End function definitions

doublefactorial = @(x) prod(3:2:x);
BiasCoeff=@(r,k,deg) multinomial(deg,[(r+1),k])*prod(fMoments([r+2 k]));

cumBLAGaussian=0; %Initialise variable holding Gaussian gain term
cumVecBLAarb=zeros(1,l_linearity);

for deg=vecDeg %For each nonlinearity degree
 if deg==1; %Linear kernel
     cumVecBLAarb=cumVecBLAarb+ones(1,l_linearity)*Pn_rev(2);
 elseif (mod(deg,2)) %Only perform calculation on odd degrees
 alphag=sum(g_linearity.^2)^((deg-1)/2); %Gaussian gain = pairwise sum
 alphag=alphag*doublefactorial(deg);
 cumBLAGaussian=cumBLAGaussian+alphag;

    cellPartitions=intpartitions((deg-1)/2); %Identify all possible ways of
        partitioning the Volterra kernel
    vecStdDscrp=zeros(1,l_linearity); %Standard discrepancy value for a particular
         NL degree

    for k=1:length(cellPartitions)
       %For degree 7, integer partition of (7-1)/2, = 3 => e.g. {1,1,2}
       vecPart = cellPartitions{k}*2; %{1,1,2} -> %{2,2,4}
     %e.g. {2,2,4} represents h(a,a,b,b,r,r,r)M_2 M_2 M_4 or h(r,a,a,b,b,b,b)M_2
         M_2 M_4
       NvecPart=length(vecPart);
       vecPartUnique = unique(vecPart);

       NtempTrunc=NvecPart-1;
       for j=1:length(vecPartUnique)  %For each unique combo, split into g(r)
           parts
     %Using above example of integer partition result {2,2,4}, unique combo is 2
         and 4.
```

```
    %In essence we are testing for both unique case: h(a,a,b,b,r,r,r)M_2 M_2 M_4,
        and h(r,a,a,b,b,b,b)M_2 M_2 M_4
          tempTrunc = vecPart;
          tempTrunc(find(vecPart==vecPartUnique(j),1,'first'))=[]; %Take first
              element

          if ~isempty(tempTrunc)
        matPower_g=zeros(l_linearity,NtempTrunc);
        for l=1:NtempTrunc;
          matPower_g(:,l)=g_linearity.^tempTrunc(l); %Raise IRF to each power
        end
        %Combinatorial correction factor needed to normalise result from
            cumprodapdci function later:
        correctionFactor = prod(factorial(histc(tempTrunc,unique(tempTrunc))));
        %cumprodapdci function implements equation A.2.1
        tempDscrp = g_linearity.^vecPartUnique(j).*(BiasCoeff(vecPartUnique(j),
            tempTrunc,deg)*cumprodapdci(matPower_g,(1:l_linearity)')'/
            correctionFactor);
          else %Lone element
              tempDscrp = g_linearity.^vecPartUnique(j).*(BiasCoeff(vecPartUnique
                  (j),0,deg));
          end
          vecStdDscrp=vecStdDscrp+tempDscrp; %Include the partial sum
      end
      %In following lines, g(k) only need to be evaluated once, using example
          before, h(a,a,b,b,c,c,c)M_2 M_2 M_4
      vecCountDOF = (histc(vecPart,vecPartUnique)); %Factorials of the degrees of
          freedom, needed for normalisation with cumprodapdci
      if any(vecCountDOF>21)
         warning('Precision may be hampered by large factorial numbers')
      end
      correctionFactor = prod(factorial(vecCountDOF)); %Product of the factorials
          of the degrees of freedom, needed for normalisation with cumprodapdci
      matPower_g=zeros(l_linearity,NvecPart);
      for l=1:NvecPart;
          matPower_g(:,l)=g_linearity.^vecPart(l);
      end
      tempDscrp = (BiasCoeff(0,vecPart,deg)*cumprodapdci(matPower_g,(1:
          l_linearity)')'/correctionFactor);
      vecStdDscrp=vecStdDscrp+tempDscrp;
    end
    cumVecBLAarb=cumVecBLAarb+vecStdDscrp*Pn_rev(deg+1); %Polynomial coefficient
end %End of each NL degree

TheoryBLA_gaussian = cumBLAGaussian*g_linearity; %Gaussian BLA result
TheoryBLA_arb = g_linearity.*cumVecBLAarb; %Gaussian BLA result
end %End of each input type
```

Listing B.5: Theoretical_BLA.m

```
function SUM = cumprodapdci(dataArray,varargin)
%cumprodapdci usage: SUM = cumprodapdci(dataArray,ignoredInd,hangingMode)
%CUMmulative PRODucts across All Permutations of Distinct Column-Indices
%This function returns the cumulative sum over products of column elements
%with all permutations of unique and distinct column-indices.
%The sum excludes automatically products containing elements with common
%column index values. In addition, any user specified index values in
%ignoredInd are excluded. As all permutations of columns are summed, the
%permutation of the columns of the input dataArray is not important.
%
%For example, cumprodapdci([a,b],[1 7]) for vectors a and b is the sum of
%sum(a.*b) minus all contributions of a(2)b(2), a(3)b(3) and the user
%specified ignoredInd of 1 and 7, i.e. all products involving the elements
```

```
%a(1),a(7),b(1) and b(7) are not included in the sum.
%
% EXAMPLE APPLICATION:
%
%   For a vector K represnting the input values of a multinomial expansion
%   of (K(1)+K(2)+K(3))^n, the result can be expressed Mathematically as:
%   sum(sum(...sum(K)*K)*K)...)
%
%   Alternatively, one can build up the expansion with product
%   elements. For example, when n=4, the result is equal to the sum of
%   all forms of:
%   x^4
%    +4*(sum of all forms of a*b^3)
%    +6*(sum of all forms of a^2*b^2)
%    +12*(all combinations of the form a*b*c^2.)
%   see http://www.wolframalpha.com/input/?i=expand%28a%2Bb%2Bc%29%5E4
%
%   Hence, if one is interested at all contributions of the form a^2*b^4
%   in n=6 for example, the answer is given as the multinomial coefficient
%   6!/2!/4! * sum of all combinations of the form a^2*b^4.
%
%   This can be performed by using this routine of cumprodapdci([K.^2,K^4]),
%   where K is [a;b;c;d+...] as in the problem of (a+b+c+d+e+...)^6.
%   If one wants to ignore contributions containing terms K(3)=c and K(5)=e,
%   one may use cumprodapdci([K.^2,K^4],[3,5]).
%
%   It is important to note that the result requires normalisation with the
%   multinomial coefficient, the implementation of which is available on
%   file exchange from other authors.
%
%   If there are common column vector with common power, such is the case
%   when n = 4 and one would like to sum all instances of the form a*b*c^2.
%   The result should be normalised by multiplying the result with
%   M/(products of degree of freedoms)!, where ! is the factorial,
%   M is the multinomial coefficient, and the degree of freedom refers to
%   the number of variables with a common power. In this example, the
%   variables with a common power are a & b hence the degree of
%   freedom is two.
%
%   For a more complicated case, for example, a*b*c^4*d^4*e^6*f^6*g^6,
%   the normalisation factor should be M/2!2!3!=M/24.
%   Here, the multinomial coefficient M=(6+6+6+4+4+2)!/(6!6!6!4!4!2!).
%
% ADDITIONAL NOTES
%
%   The calculation involved is complex and the function computation time
%   is of order O(R^C) where R and C are the number of rows and columns of
%   the dataArray respectively.
%
%   If number of columns of dataArray is 2, this function gives the same
%   result as:
%   1) The sum of elements of the Dyadic (Tensor or outer) product of column
%      1 and 2 minus the trace of the Dyadic product matrix. In MATLAB the
%      'kron' function performs the Dyadic product.
%   2) The circular cross-correlation of column 1 and column 2 minus the
%      zero shift value.
%%
Rows = size(dataArray,1);
Cols = size(dataArray,2);
flagHanging=false;
switch nargin
    case 1
    ignoredInd={[]};
    N=1; %For-loop compatibility
```

```
78          if Cols == 0 || Rows == 0
               SUM=[]; return;
80          end
            if Cols > Rows
82             warning('cumprodapdci received an input array with more columns than
                   rows, possibly unintended input orientation')
               SUM=0; return;
84          end
        case 2
86
            if isempty(varargin{1})
88              ignoredInd=[];
                N=1; %For-loop compatibility
90          else
                if ~isa(varargin{1}, 'cell')
92                  temp = varargin{1};
                    N = size(temp,1);
94                  %Pre-condition and convert to cell array
                    temp(temp<=0)=NaN;
96                  temp(temp>Rows)=NaN;
                    ignoredInd=num2cell(temp,2);
98                  ignoredInd=ignoredInd(:);
                else %For cell input, orientation does not matter
100                 ignoredInd=varargin{1};
                    if isvector(ignoredInd)
102                     N = length(ignoredInd);
                    else
104                    error('ignoredInd as should be a cell vector of dimension Nx1')
                    end
106             end
                for ig = 1:N
108                 ignoredInd{ig} = ignoredInd{ig}(~isnan(ignoredInd{ig}));
                end
110         end
            if Cols == 0 || Rows == 0
112            SUM=[]; return;
            end
114         if Cols > Rows
               warning('CUMPRODNEI receiced an input array with more columns than rows
                     , possibly unintended input orientation')
116            SUM=0; return;
            end
118     case 3 %Hanging mode for nested call, i.e. nested function call
            N=Rows;
120         flagHanging=true;
        otherwise
122         error('Too many input arguments')
    end
124 SUM=zeros(N,1); %Pre-allocation
    col1Sum = sum(dataArray(:,1)); %stored sum for column 1, code efficiency purposes
126 switch Cols
        case 1 %Single column data
128
            if flagHanging %Hanging mode for nested call
130             SUM=col1Sum-dataArray;
            else
132             for ig = 1:N
                    SUM(ig)=sum(dataArray(ignoredInd{ig}));
134             end
                SUM=col1Sum-SUM; %Sum all elements and take away those unwanted.
136         end
            return
138     case 2 %Two columns
            if flagHanging %Hanging mode for nested call, slightly faster code path
```

```
                    [X,Y]=meshgrid(col1Sum-dataArray(:,1),dataArray(:,2));
                    temp=Y;
                    temp(1:N+1:end)=0; %Zero the diagonals
                    SUM=sum(X.*temp-temp.^2).';
            else
                for ig = 1:N
                    temp=dataArray;
                    temp(ignoredInd{ig},:)=0;
                    SUM(ig)=sum((col1Sum-sum(dataArray(ig,1))-temp(:,1)).*temp(:,2));
                        %(Sum A - A) .* B
                end
            end
            return
        otherwise %Three or more columns
            if flagHanging %Hanging mode for nested call, slightly faster code path
                for ig = 1:N %
                    temp=dataArray;
                    temp(ig,:)=0;
                    SUM(ig)=sum(cumprodapdci(temp(:,1:Cols-1),[],true).*temp(:,Cols));
                end
            else
                for ig = 1:N
                    temp=dataArray;
                    temp(ignoredInd{ig},:)=0;
                    SUM(ig)=sum(cumprodapdci(temp(:,1:Cols-1),[],true).*temp(:,Cols));
                end
            end
    end
    end %End function
```

Listing B.6: cumprodapdci.m

```
function cellPart = intpartitions(intIn)
%INTPARTITION performs integer partition, i.e. the partition of of a set
%containing homogenous elements. The function generates a cell array
%containing a list of vectors representing all possible ways
%of partitioning a set containing intIn number of identical elements
%without order awareness. The numerical representation in the
%output describes the partitions as: {[3 1]} = [1 1 1 | 1]
%   Example output: intpartition(4) gives {[1 1 1 1];[2 1 1];[2 2];[3 1];4}
%   Number of ways of partitioning is according to sequence:
%   http://oeis.org/A000041 - (1), 1, 2, 3, 5, 7, 11, 15, 22, 30, 42, ...
intIn = round(intIn);
if ~isscalar(intIn)
    error('Invalid input. Input must be a scalar integer')
end
vecWorking = ones(1,intIn);
cellPart(1) = {vecWorking};
while vecWorking(1)<intIn;
    flag1 = true;
    indPrime=1;
    while flag1
        %Not yet finished for the this group
        if length(vecWorking)>2
            %Combine unit elements
            indj = find(vecWorking==1,1,'first');
            if ~isempty(indj)
                while (indj <= length(vecWorking)-1) && (vecWorking(indj) == 1)
                    newDigit = vecWorking(indj)+1;
                    vecTempFront = vecWorking(1:indj-1);
                    vecTempEnd = ones(1,intIn-newDigit-sum(vecTempFront));
                    vecWorking = [vecTempFront,newDigit,vecTempEnd];
                    cellPart(end+1) = {vecWorking};
```

```
                    indj=indj+1;
33                end
            end
35      end
        if indPrime >= length(vecWorking)
37          indPrime=length(vecWorking)-1; %Prevent overflow
        end
39      if vecWorking(indPrime+1)+1 <= vecWorking(indPrime) &&...
                (sum(vecWorking(1:indPrime+1))<intIn)
41          indPrime = indPrime+1;
            vecWorking(indPrime)=vecWorking(indPrime)+1;
43          vecTempFront = vecWorking(1:indPrime);
            vecTempEnd = ones(1,intIn-sum(vecTempFront));
45          vecWorking = [vecTempFront,vecTempEnd];
            cellPart(end+1) = {vecWorking};
47      else
            indPrime=indPrime-1; %Back stepping
49          if indPrime < 1
                flag1 = false;
51          end
        end
53  end
    vecTempFront=vecWorking(1)+1;
55  vecTempEnd=ones(1,intIn-vecTempFront);
    vecWorking = [vecTempFront,vecTempEnd];
57  cellPart(end+1) = {vecWorking};
end
59 cellPart=cellPart(:); %Output a column cell
end
```

Listing B.7: intpartitions.m